Embedded Systems

Week 1: Introduction

Dr. Vecdi Emre Levent

Instructors

Assist. Prof. Dr. Vecdi Emre Levent

Email : emre@levent.tc

emre.levent@marmara.edu.tr

Web: www.levent.tc

Course Plan

- Introduction
  - Embedded Systems and Applications

"Embedded" System

- Integrated into the Device or System
- Executes Specific (Not General Purpose) Tasks
- Generally Invisible to Users
  - Refrigerators (Temperature Control)
  - Washing Machines (Cycle Management)
  - Microwave Ovens (Timer and Power Control)
  - Dishwashers (Water Flow Control)
  - Smart TV (Signal Processing)

Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Depends on Requirements of Application

Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

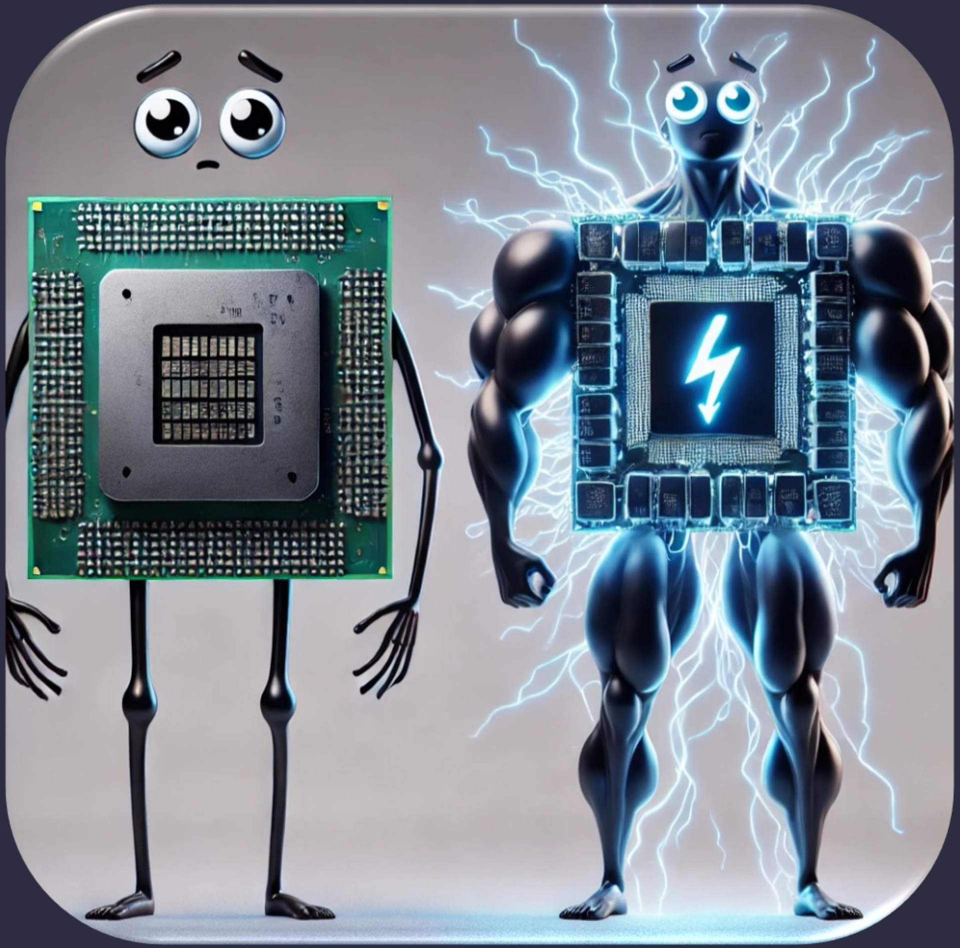
- <u>Depends on Requirements of Application</u>
  - Budget
    - Production Cost Target
      - Component Selection
      - Team Building

Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Depends on Requirements of Application
  - Power Consumption



Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

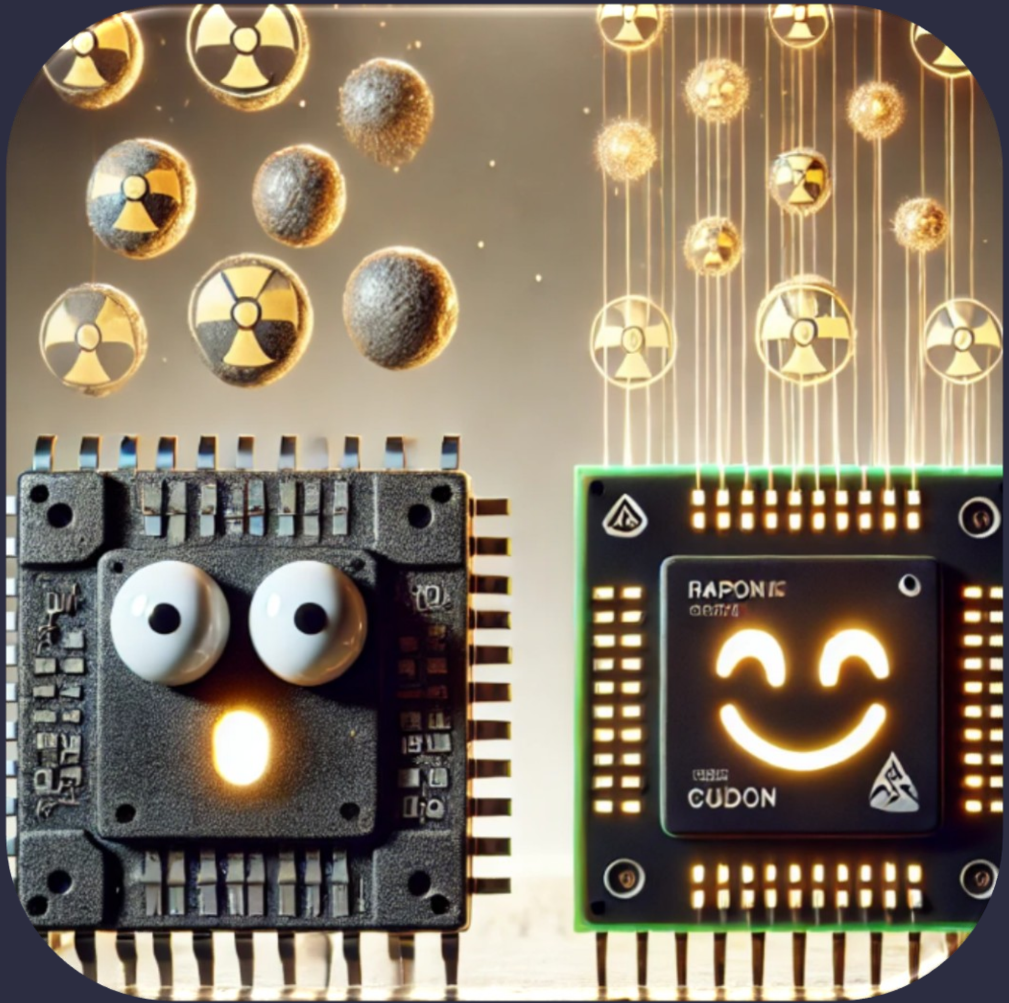
- Depends on Requirements of Application
  - Thermal Requirements

Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- <u>Depends on Requirements of Application</u>
  - Radiation Immunity Requirements (Aviation and Space Grade)



Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

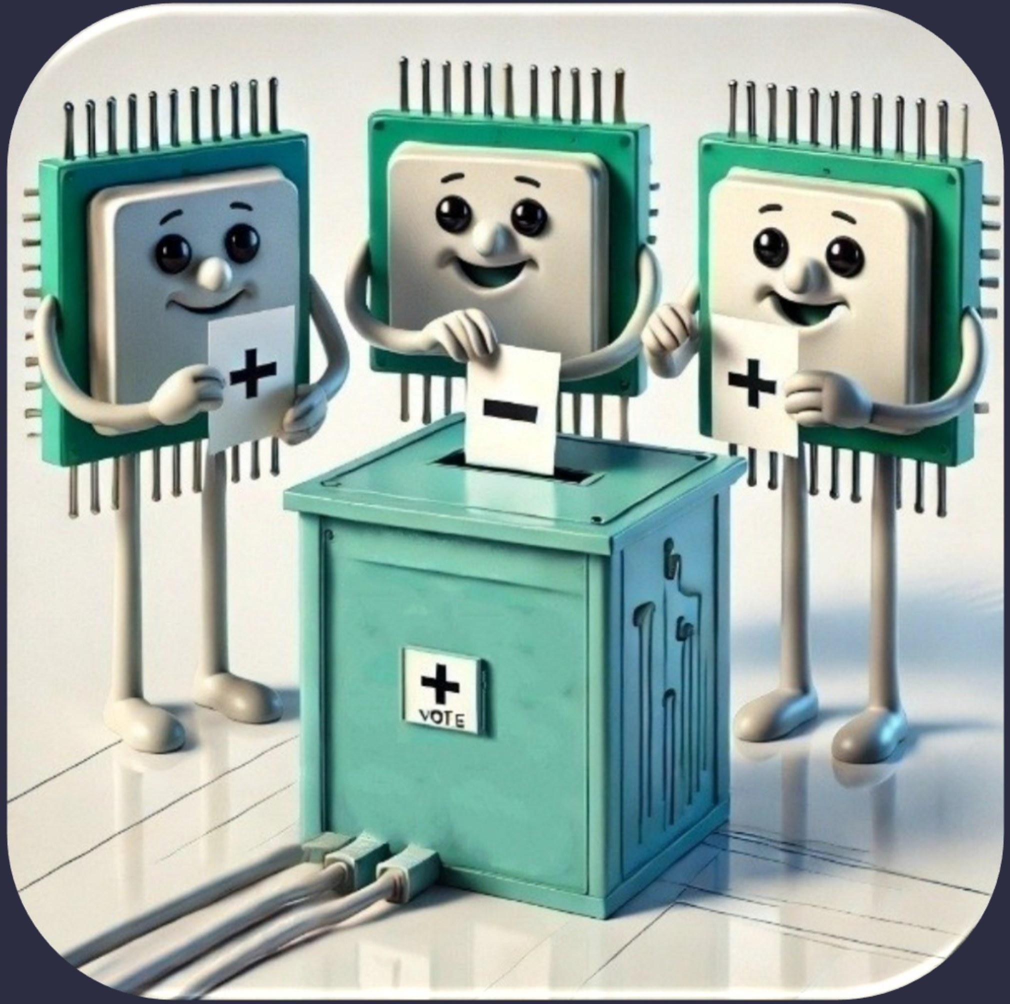
- <u>Depends on Requirements of Application</u>
  - Safety Requirements
    - Triple Modular Redundancy
    - Error Detection/Correction
      - Hamming Codes
      - Error Correcting Codes

Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Depends on Requirements of Application
  - Safety Requirements
    - Triple Modular Redundancy



Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

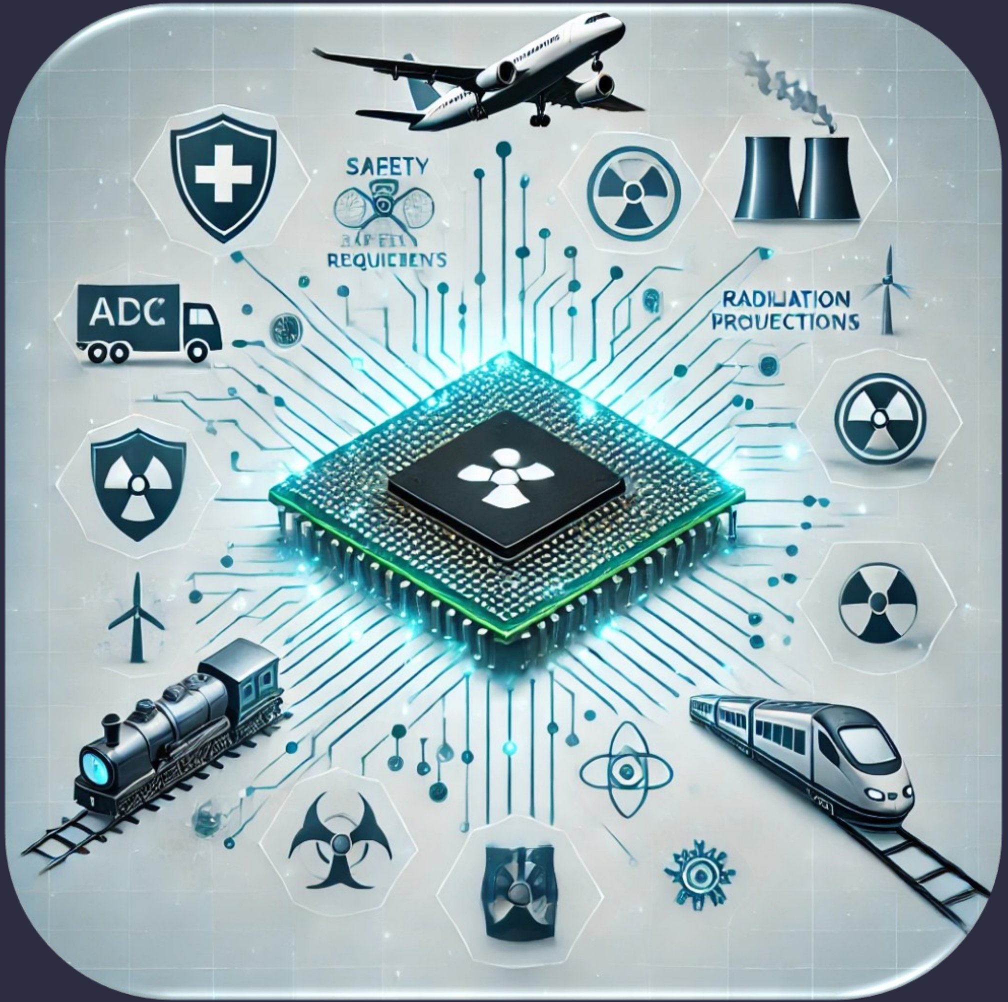
- <u>Depends on Requirements of Application</u>
  - Safety Requirements
    - Error Detection/Correction
      - Hamming Codes
      - Error Correcting Codes

Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Depends on Requirements of Application
  - Standards
  - Aerospace and Avionics
    - DO-254 – Design assurance for airborne electronic hardware (FAA, EASA)
    - DO-178C – Software considerations in airborne systems (FAA, EASA)
    - ARP 4754A – Guidelines for aircraft and system development
    - ARP 4761 – Safety assessment process for civil airborne systems

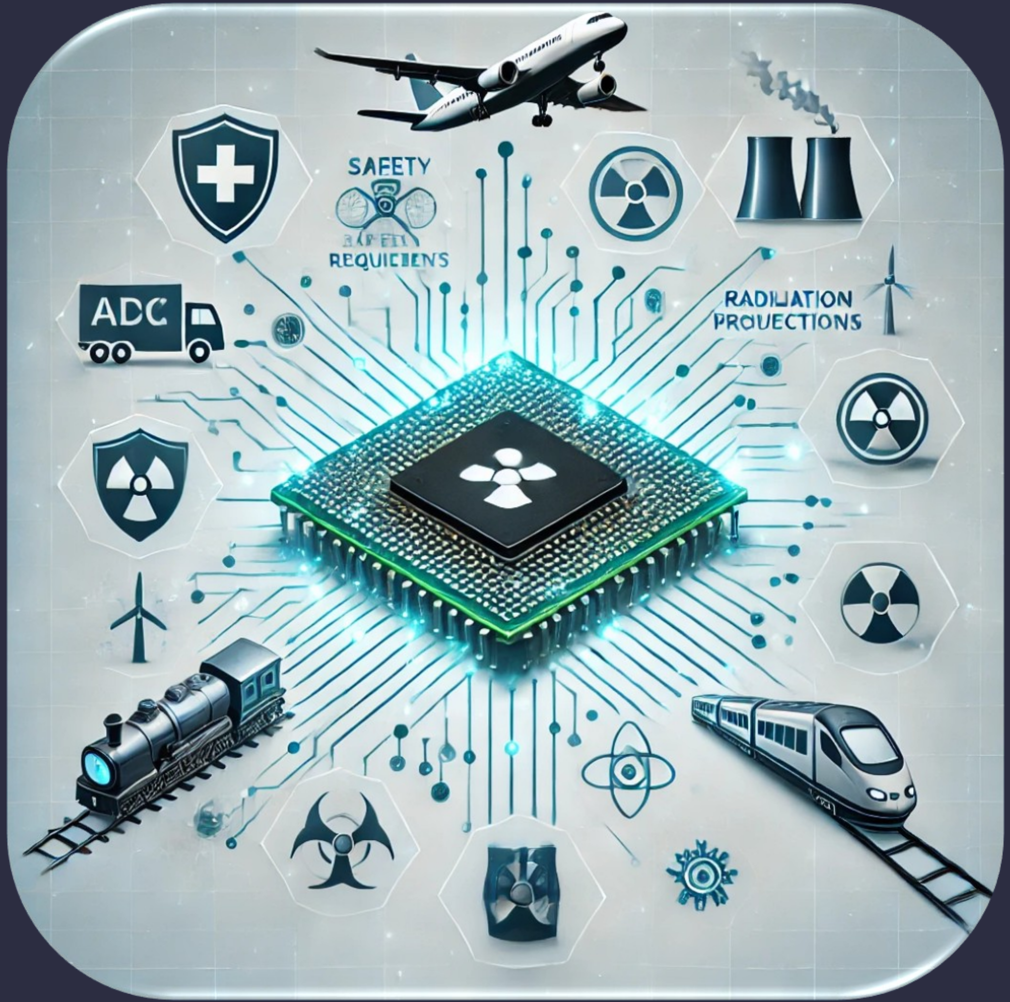


Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- <u>Depends on Requirements of Application</u>
  - Standards
  - Automotive
    - ISO 26262 – Functional safety for road vehicles
    - SAE J3061 – Cybersecurity guidebook for automotive systems
    - ISO/SAE 21434 – Automotive cybersecurity engineering

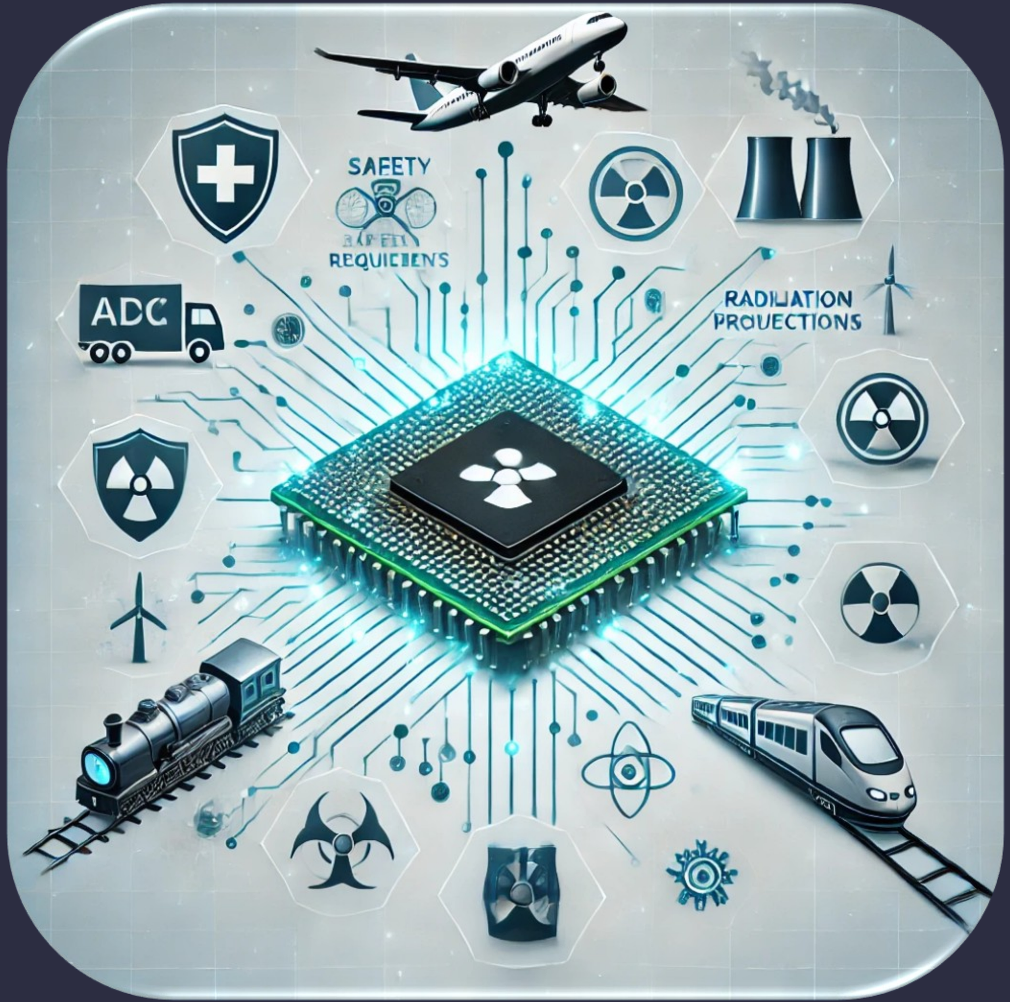


Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- <u>Depends on Requirements of Application</u>
  - Standards
  - Railway
    - EN 50126 – Reliability, availability, maintainability, and safety (RAMS) for railway applications
    - EN 50128 – Safety-related software development for railway control systems
    - EN 50129 – Railway signaling system safety requirements

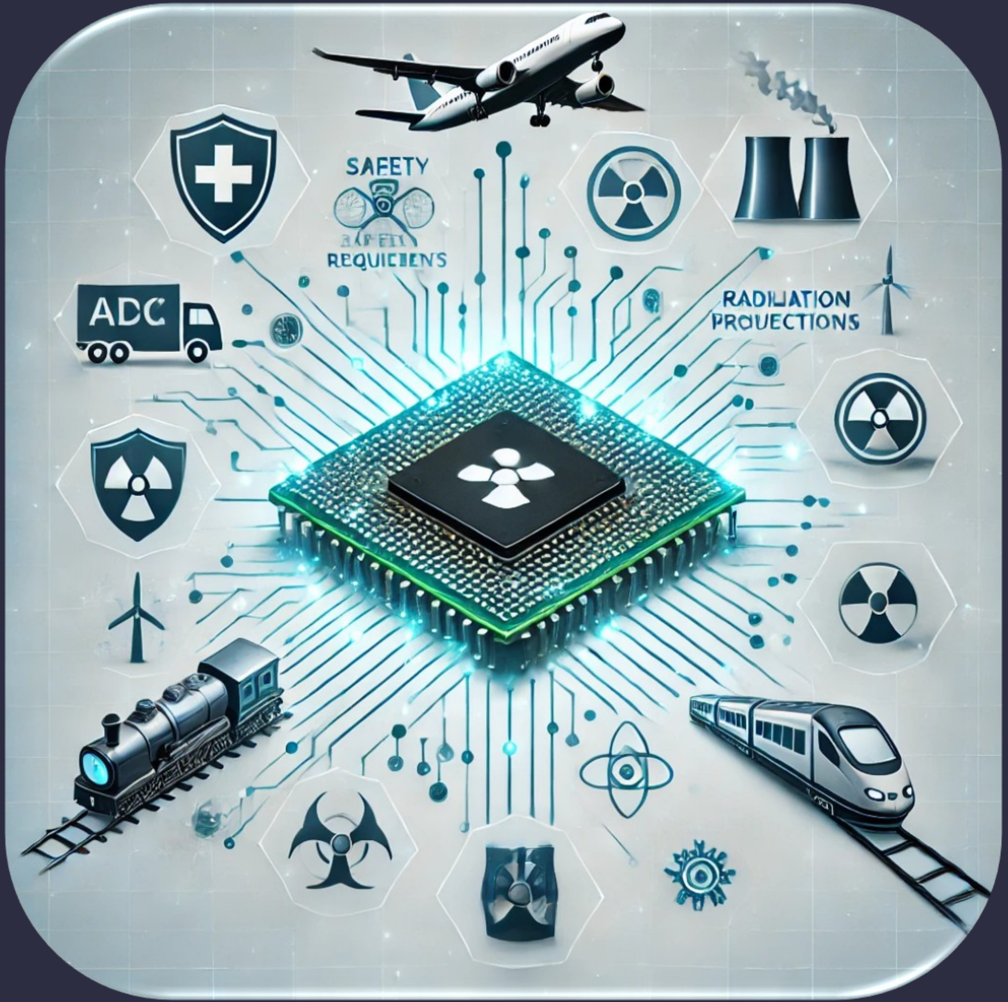


Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Depends on Requirements of Application
  - Standards
  - Medical Devices
    - IEC 60601 – Safety requirements for medical electrical equipment
    - ISO 14971 – Risk management for medical devices
    - IEC 62304 – Software lifecycle processes for medical device software

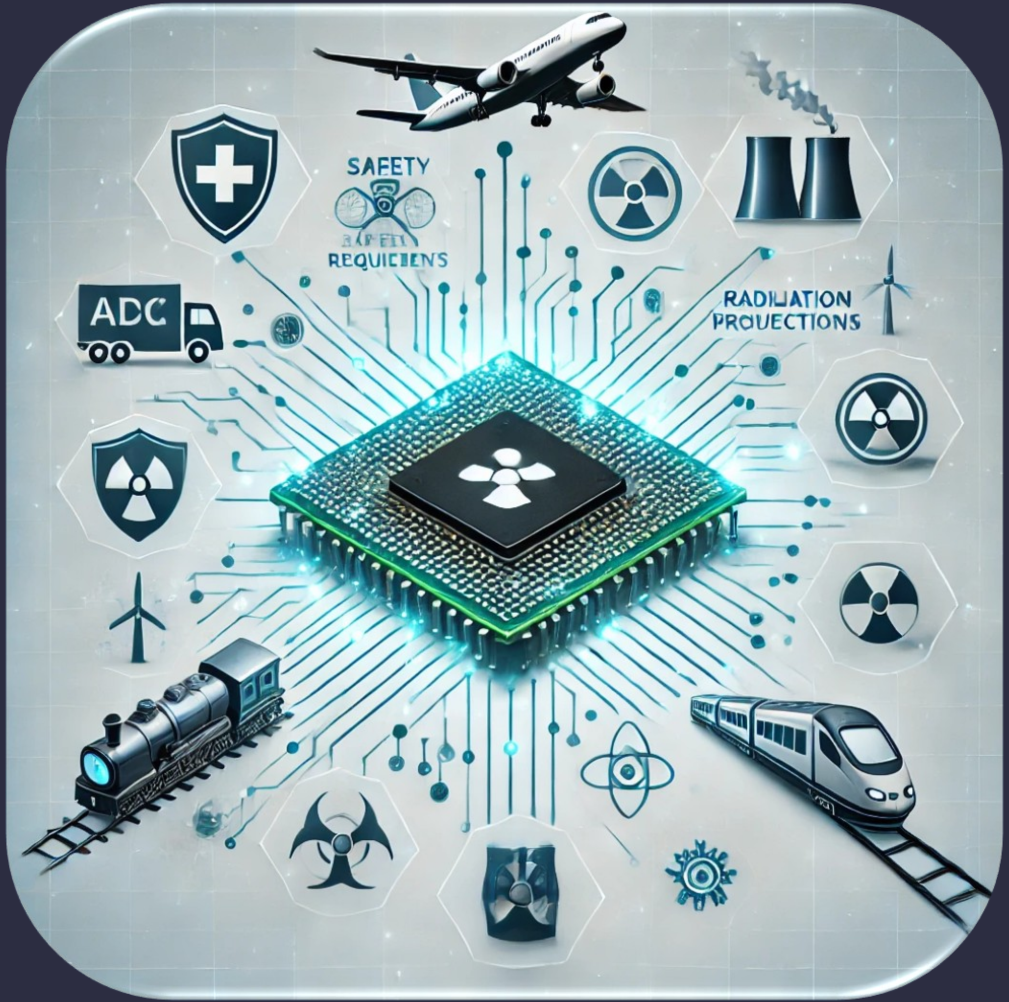


Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- <u>Depends on Requirements of Application</u>
  - Standards
  - Nuclear and Energy Sector
    - IEC 61513 – Safety requirements for nuclear power plant instrumentation and control systems
    - IEEE 7-4.3.2 – Criteria for digital computers in safety systems of nuclear power plants

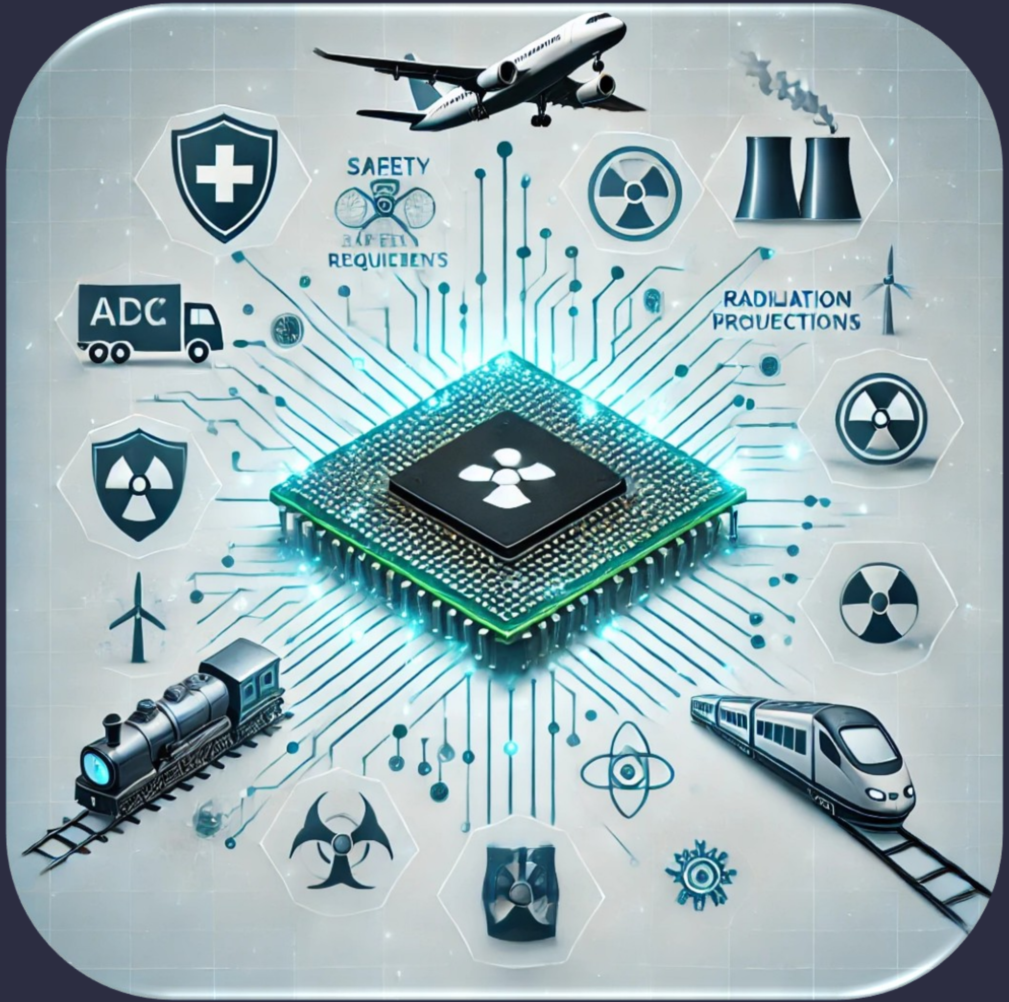


Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- <u>Depends on Requirements of Application</u>
  - Standards
  - Defense and Military Systems
    - MIL-STD-882E – System safety management for defense applications
    - NIST SP 800-53 – Cybersecurity and risk management framework
    - ISO/IEC 15408 (Common Criteria) – Security evaluation for IT products



- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Buttons
- Sensors
  - Temperature
  - IMU
  - GPS
  - …
- Communication Interfaces
  - UART
  - SPI
  - I2C
  - Ethernet
  - …
- RF Transceiver

- LEDs
- Motors
- Communication Interfaces
  - UART
  - SPI
  - I2C
  - Ethernet
  - …
- RF Transceiver

Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options

- MCU (Microcontroller Unit)
- MPU (Microprocessor Unit)
- CPU (Central Processing Unit)
- GPU (Graphics Processing Unit)
- TPU (Tensor Processing Unit)
- FPGA (Field Programmable Gate Arrays)
- ASIC (Application-specific Integrated Circuits)

Course Plan

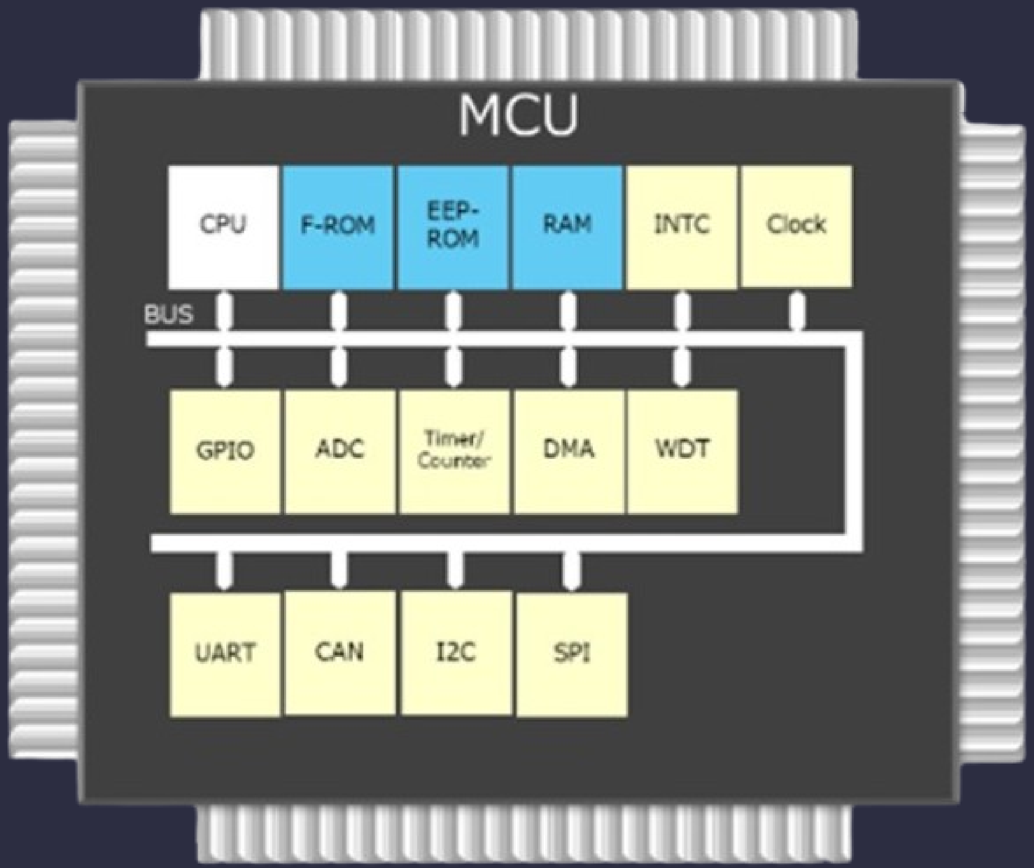
- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options

- MCU (Microcontroller Unit)

  - A low-power, single-chip processor optimized for real-time control.
  - Typically includes built-in RAM and Flash memory, no external storage required.
  - Handles simple tasks, does not perform complex computations.
  - Average Frequency: 1 MHz - 500 MHz
  - Average FLOPS: 10 MFLOPS - 100 MFLOPS



Course Plan

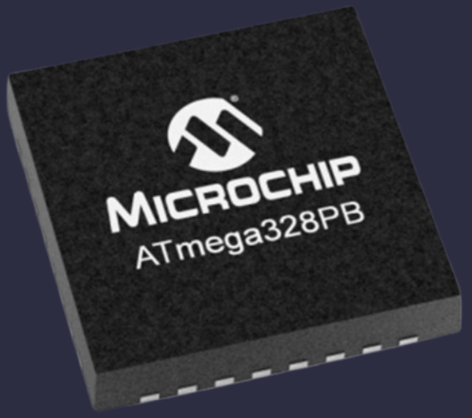
- Introduction
    - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options

- MCU (Microcontroller Unit)

- Vendors
    - Microchip Technology (PIC, AVR)
    - ST Microelectronic (STM8)
    - Texas Instruments (MSP430)
    - Renesas Electronics (RL78)

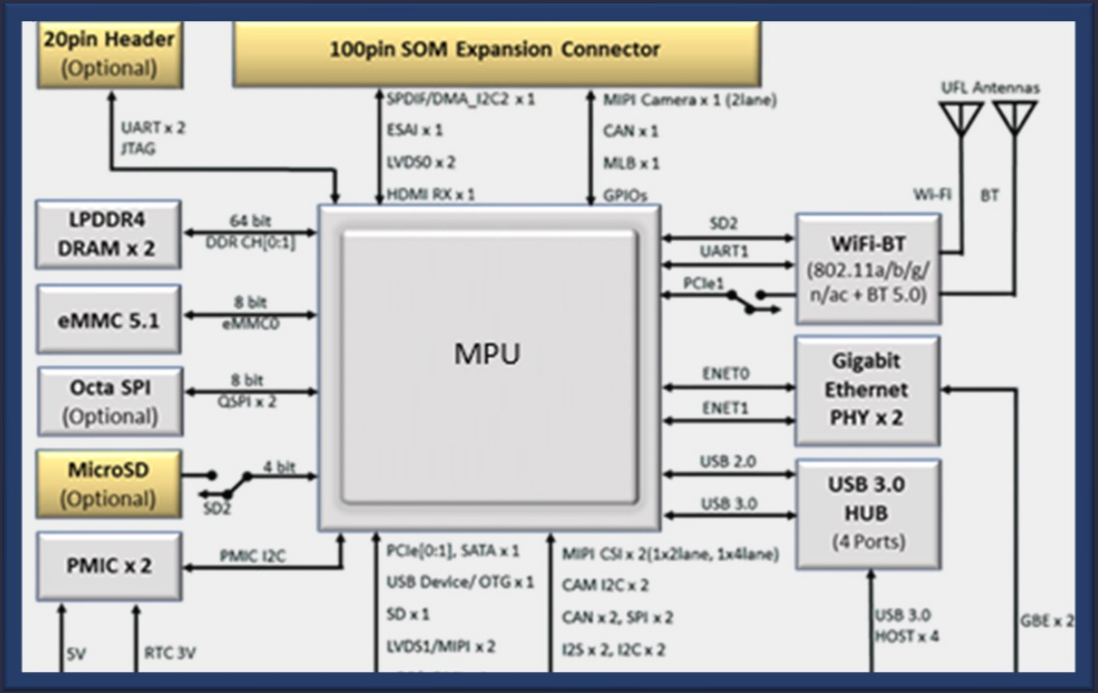


Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options
- MPU (Microprocessor Unit)
  - Works with external RAM and storage, often used in embedded Linux systems.
  - More powerful than an MCU but not as strong as a CPU.
  - Handles higher-performance tasks requiring an OS.
  - Average Frequency: 500 MHz - 2.5 GHz
  - Average FLOPS: 1 GFLOPS - 10 GFLOPS



Course Plan

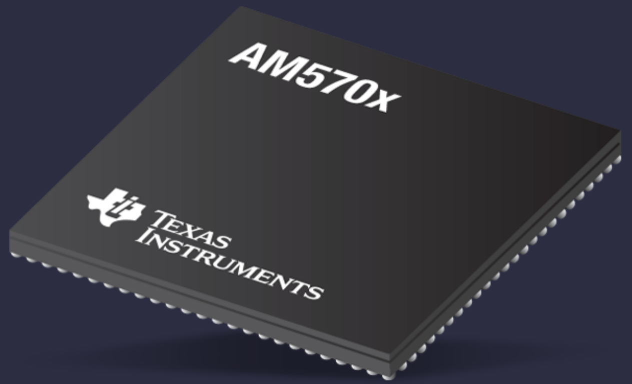
- Introduction
  - Embedded Systems and Applications

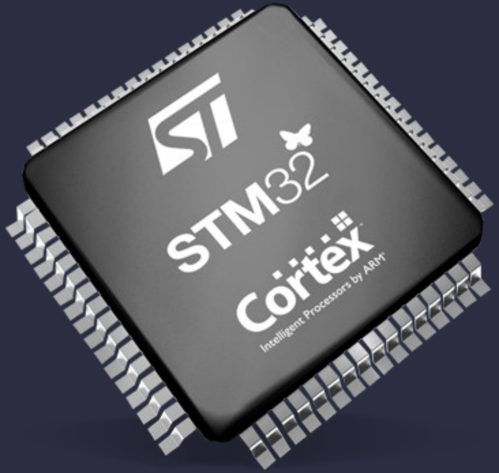
How to Build an Embedded System?

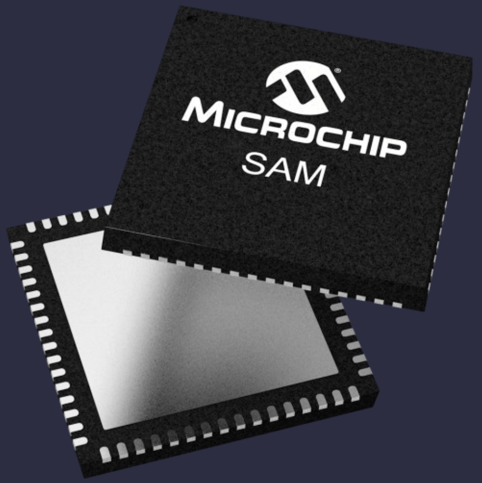
- Controller Hardware Options

- MPU (Microprocessor Unit)

- Vendors
  - NXP Semiconductors (i.MX)
  - Texas Instruments (Sitara)
  - ST Microelectronics (STM32)
  - Microchip Technology (SAM)







Course Plan

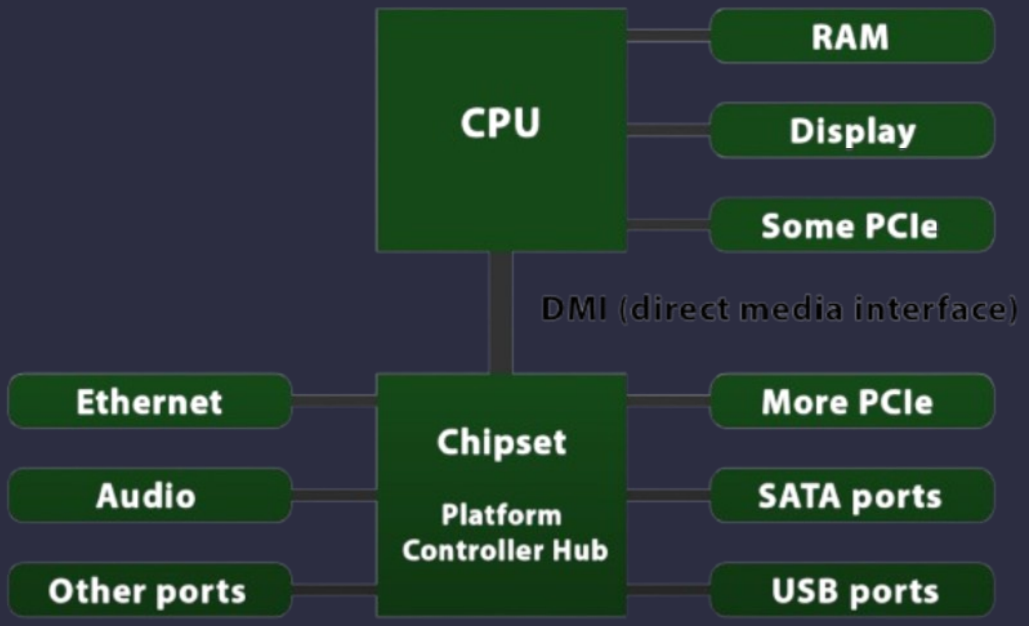
- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options

- CPU (Central Processing Unit)

  - A general-purpose processor used in computers, servers, and workstations.
  - Optimized for sequential tasks and general computing operations.
  - Has multiple cores, high clock speeds (GHz), and large cache memory.
  - Supports various operating systems (Windows, Linux, macOS).
  - Average Frequency: 1.5 GHz - 5.5 GHz
  - Average FLOPS: 100 GFLOPS - 2 TFLOPS



Course Plan

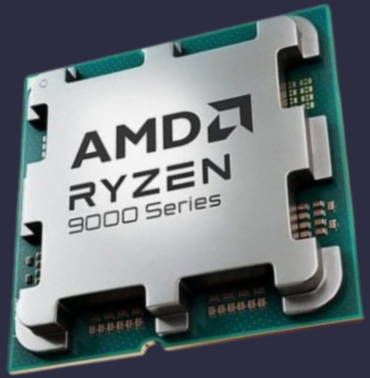
- Introduction
  - Embedded Systems and Applications

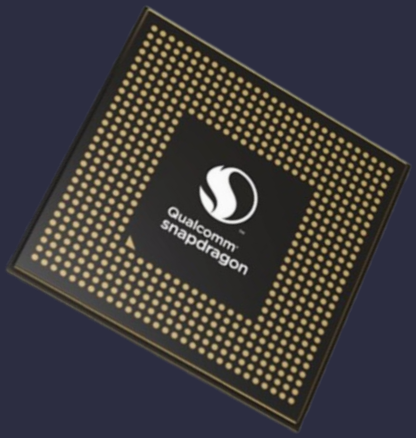
How to Build an Embedded System?

- Controller Hardware Options

- CPU (Central Processing Unit)

- Vendors
  - Intel (I9 CPU)
  - AMD (Ryzen)
  - Apple (M1, M2, M3)
  - Qualcomm (Snapdragon)



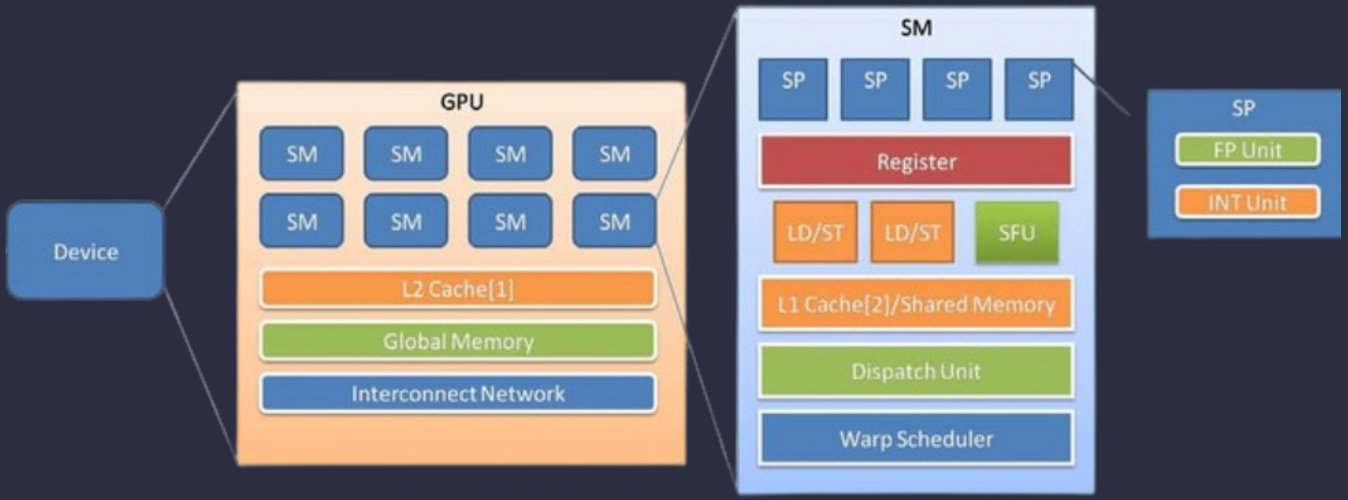


Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options
- GPU (Graphics Processing Unit)
  - Designed for parallel processing, specialized in graphics rendering and AI workloads.
  - Thousands of cores enable massive parallel execution.
  - Used in gaming, deep learning, and scientific simulations.
  - Average Frequency: 500 MHz - 2.5 GHz
  - Average FLOPS: 5 TFLOPS - 80 TFLOPS



Course Plan

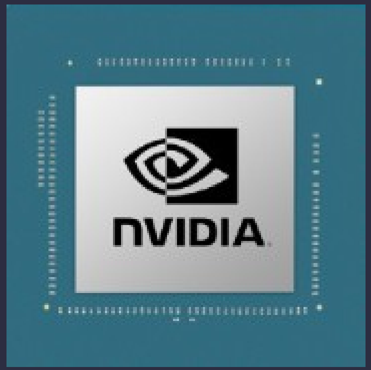
- Introduction
  - Embedded Systems and Applications

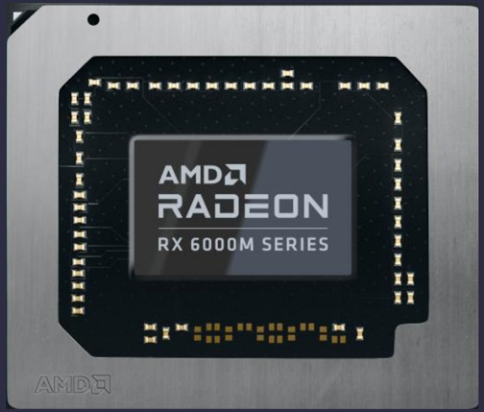
How to Build an Embedded System?

- Controller Hardware Options

- GPU (Graphics Processing Unit)

- Vendors
  - NVIDIA(Orin)
  - AMD (Radeon RX)
  - Intel (ARC A770)
  - Apple (A17)



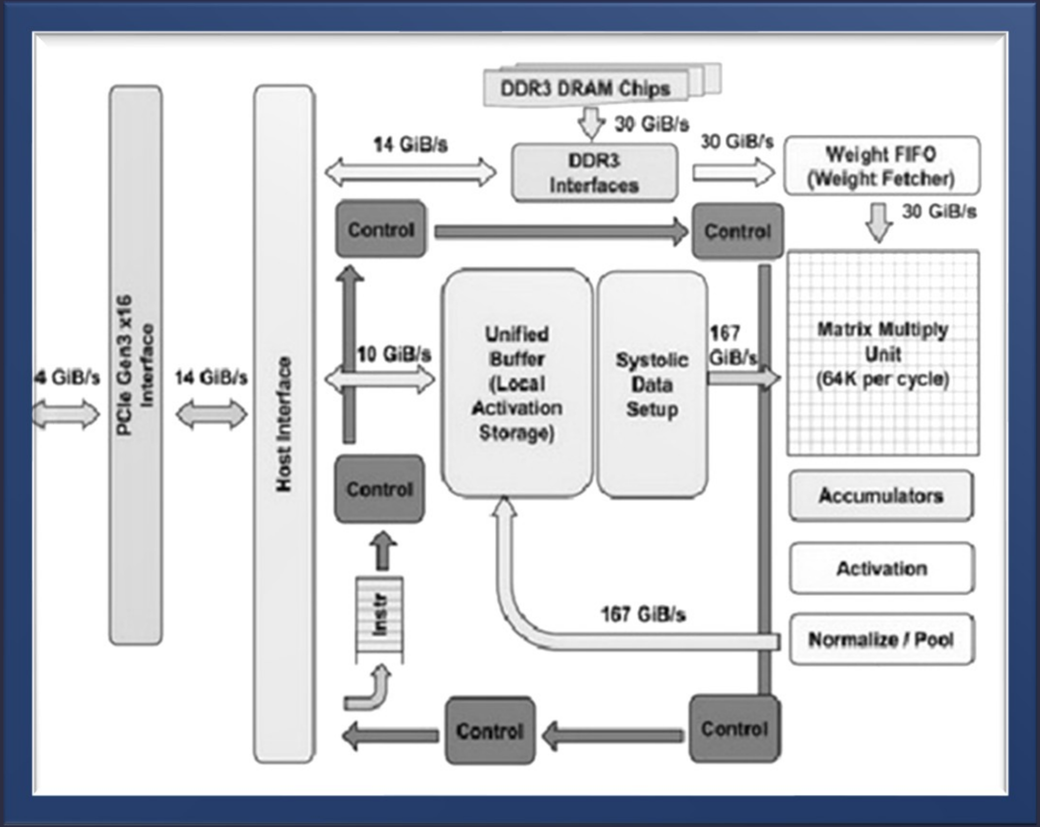


Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options
- TPU (Tensor Processing Unit)
  - A specialized processor developed by Google for AI and deep learning applications.
  - Optimized for matrix and tensor operations, used for neural network acceleration.
  - Consumes less power than GPUs for AI workloads.
  - Average Frequency: 700 MHz - 2 GHz
  - Average FLOPS: 30 TFLOPS - 100+ TFLOPS



Course Plan

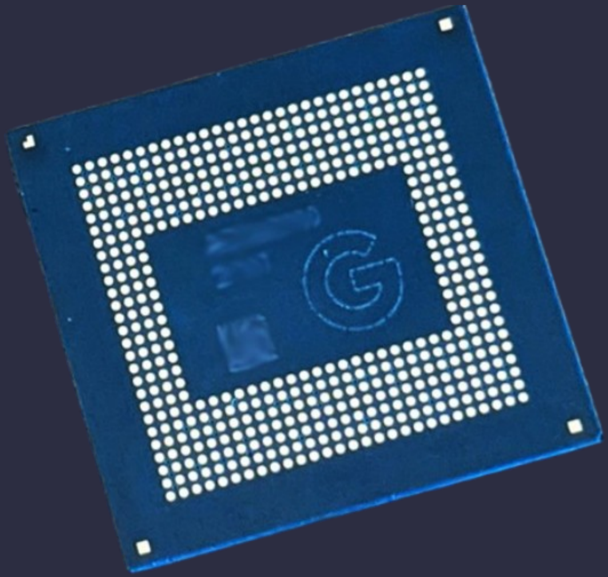
- Introduction
  - Embedded Systems and Applications

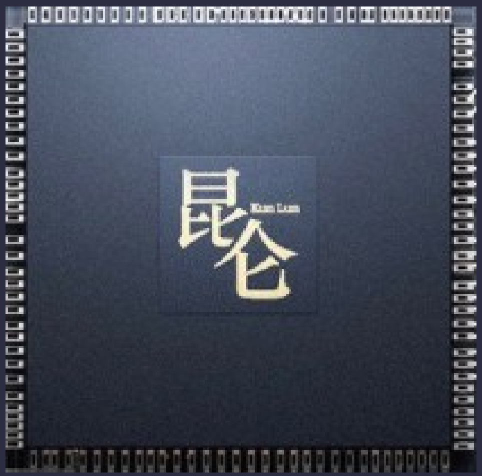
How to Build an Embedded System?

  - Controller Hardware Options

- TPU (Tensor Processing Unit)

- Vendors
  - Google (TPU v5)
  - Alibaba Cloud - Hanguang TPU
  - Baidu - Kunlun TPU





Course Plan

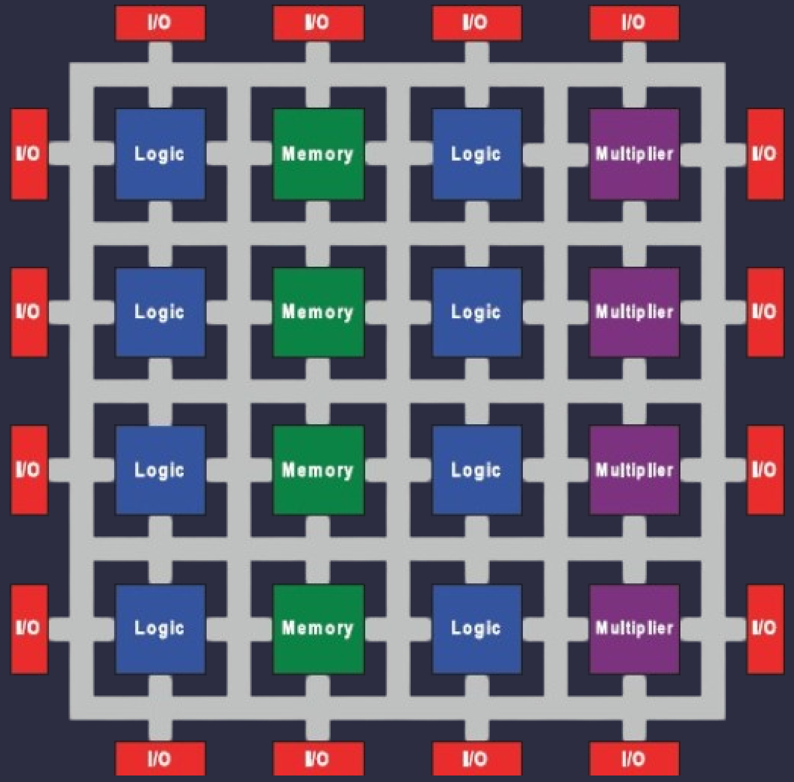
- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options

- FPGA (Field Programmable Gate Arrays)

- A reprogrammable chip that can be customized for specific hardware acceleration tasks.
- Faster than CPUs and GPUs for specialized workloads but requires FPGA Design.
- Commonly used in real-time processing and low-latency applications
- Average Frequency: 100 MHz - 1.5 GHz
- Average FLOPS: 1 TFLOPS - 10 TFLOPS (depending on design)



Course Plan

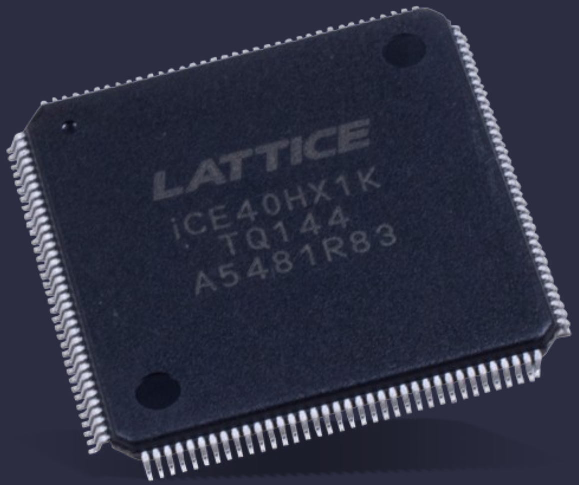
- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options

- FPGA (Field Programmable Gate Arrays)

- Vendors
  - AMD (Ultrascale+)
  - Intel (Agilex, Stratix)
  - Microchip (Polarfire)
  - Lattice Semiconductor (ECP, ICE)



Course Plan

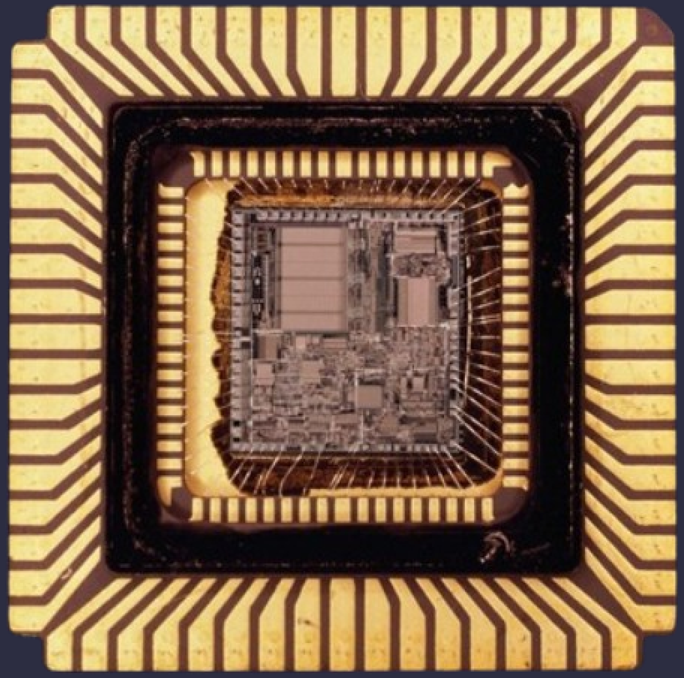
- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options

- ASIC (Application-specific Integrated Circuits)

- Fixed-function chip designed for maximum efficiency in specific applications.
- Cannot be reprogrammed like an FPGA, but offers superior power efficiency.
- Average Frequency: 500 MHz - 5 GHz
- Average FLOPS: Up to 100+ TFLOPS (depends on application)



Course Plan

- Introduction
  - Embedded Systems and Applications

How to Build an Embedded System?

- Controller Hardware Options

- ASIC (Application-specific Integrated Circuits)

- Vendors
  - TSMC
  - Samsung Foundry
  - Broadcom
  - Marvell Technology

Chip Production

What is the Chip Production and Product Transformation Process?

1

Design and Verification of Related Task in Computer Environment

Hardware Design and Verification Languages: Verilog

Design Files to Chip Manufacturing Factory

2

Transforming Silicon into a Circuit and Packaging by Using Various Light Sources and Chemical Processes

Physical Chips

3

Design of Conductive Copper Sheet (Printed Circuit Board, PCB) to Allow Chips to Transmit Signals to Each Other in a Computer Environment

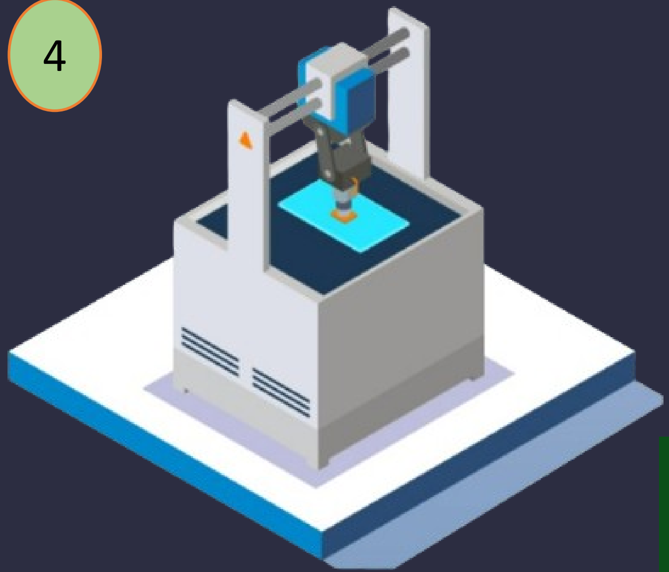
Design Files to PCB Manufacturing Factory

Chip Production

What is the Chip Production and Product Transformation Process?

4

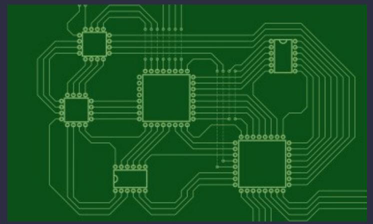
Design Files to PCB Manufacturing Factory

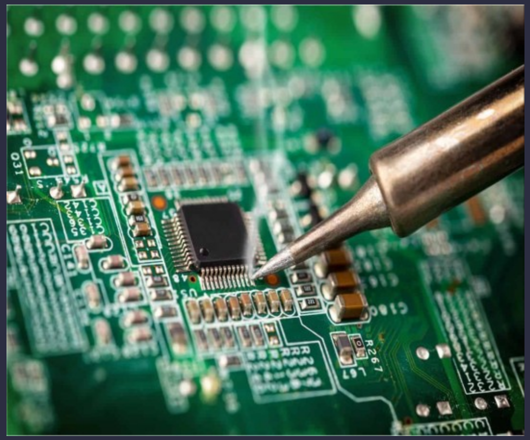


PCB Manufacture

5

PCB

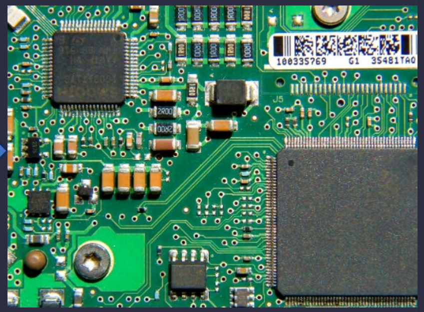




PCB Placement

6

Product

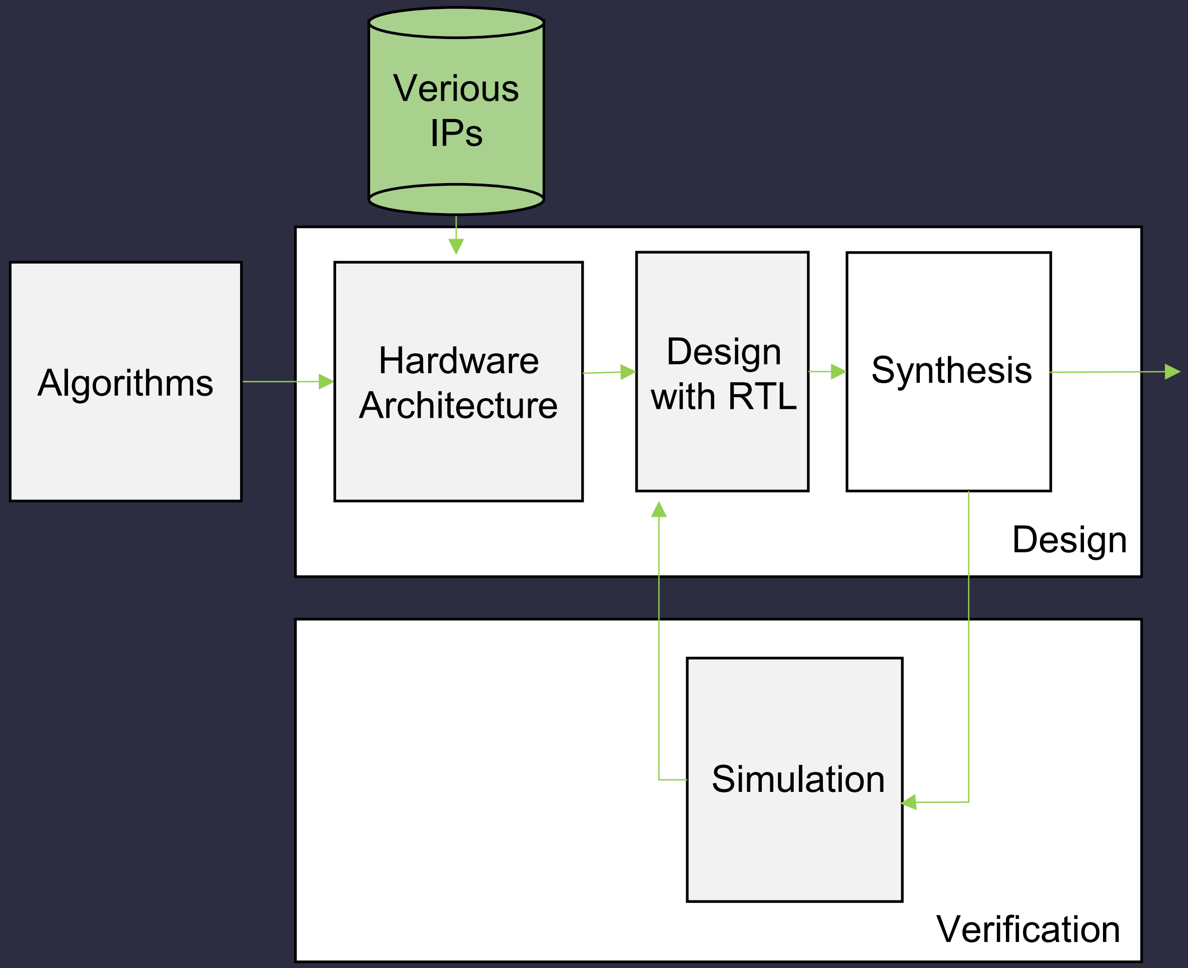


Chip Production

What is the Chip Production and Product Transformation Process?

Design and Verification of Related Task in Computer Environment

Hardware Design and Verification Languages: Verilog



Chip Production

What is the Chip Production and Product Transformation Process?

1

Design and Verification of Related Task in Computer Environment

Hardware Design and Verification Languages: Verilog

Ariane 5 (1996)

It fell due to an error made by the insufficiently verified account unit. Half a billion dollars in losses!

Chip Production

What is the Chip Production and Product Transformation Process?

2

Transforming Silicon into a Circuit and Packaging by Using Various Light Sources and Chemical Processes

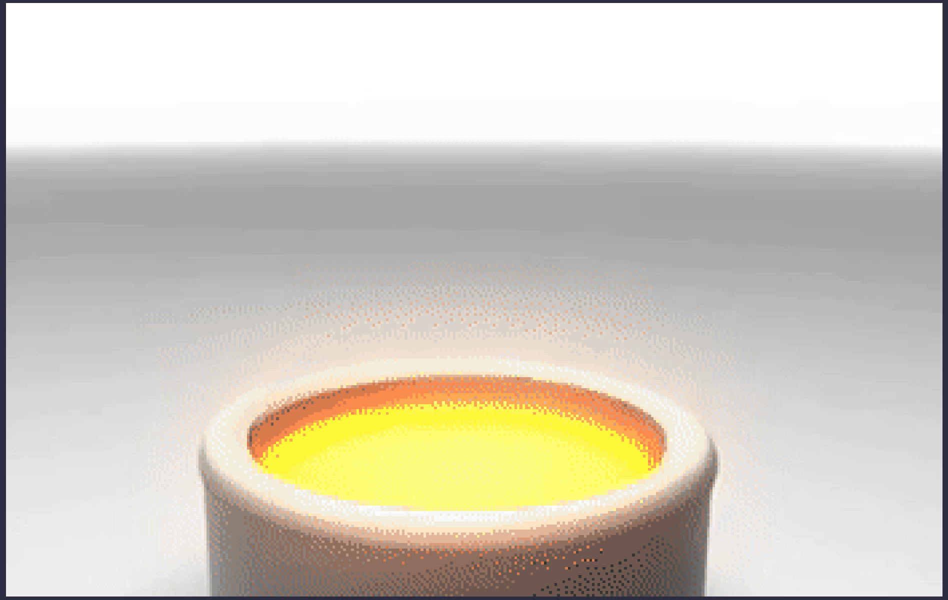
Obtaining silicon dioxide is obtained from beach sands. It is purified by chemical reaction with carbon and hydrogen gas at high temperatures (2000 ° C). Silicon Metal is obtained.

Chip Production

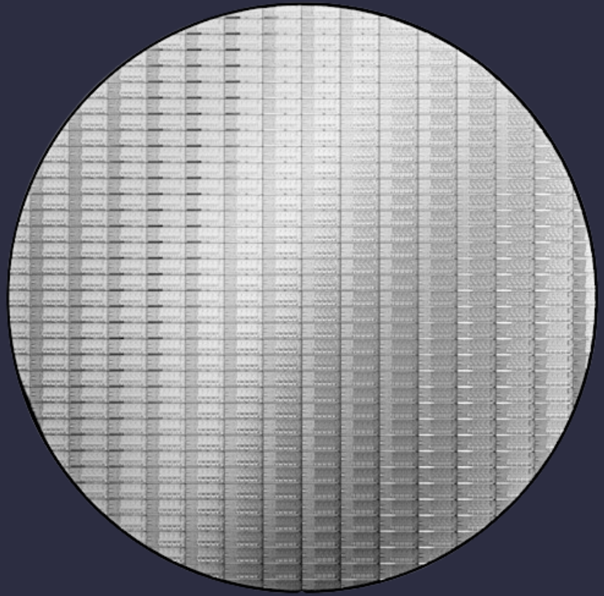
What is the Chip Production and Product Transformation Process?

2

Transforming Silicon into a Circuit and Packaging by Using Various Light Sources and Chemical Processes



Silicon metal is cut into thin disks called wafers, which will be the base material for the chips.

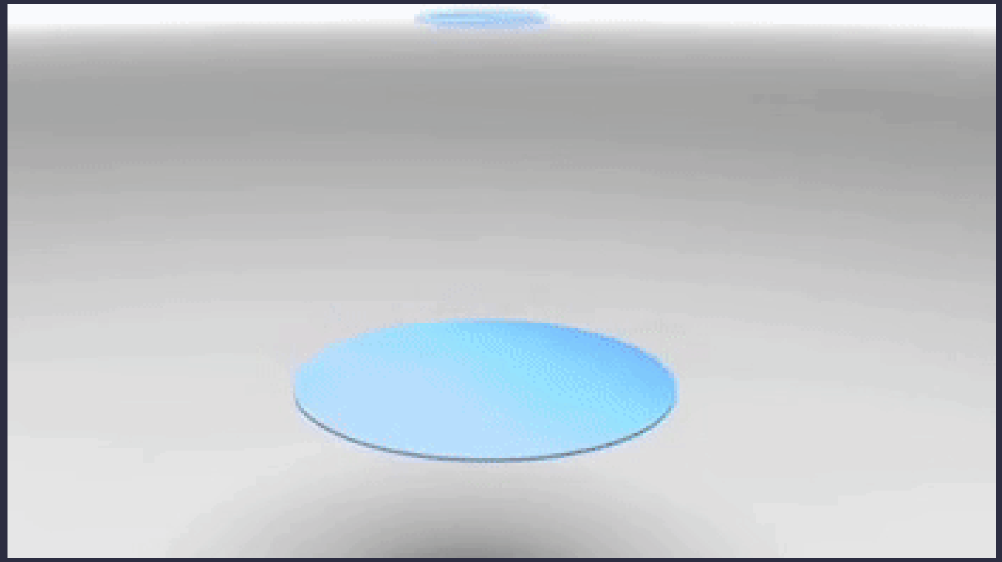


Chip Wafer

Chip Production

What is the Chip Production and Product Transformation Process?

Transforming Silicon into a Circuit and Packaging by Using Various Light Sources and Chemical Processes



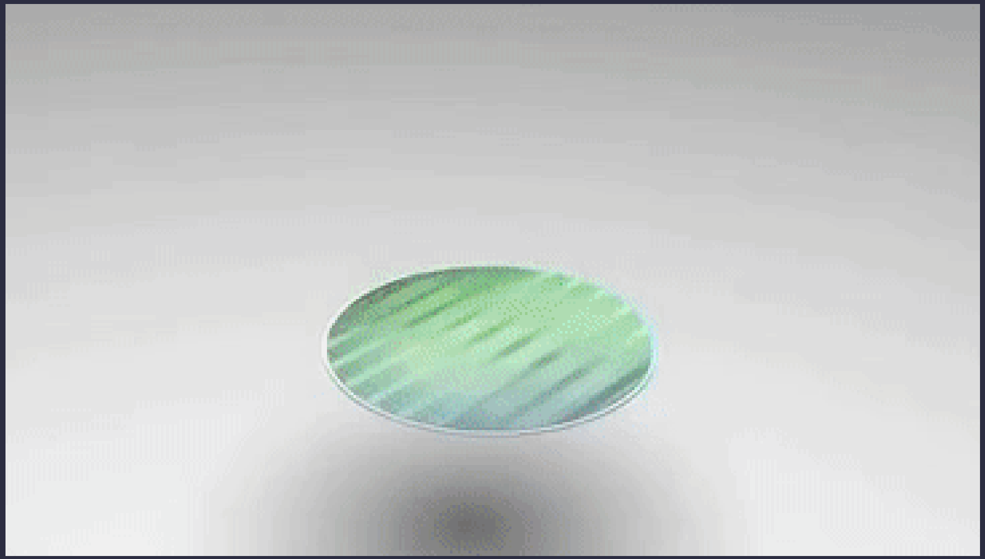
Photolithography processes are applied to the wafer, which define the chip's pattern. This pattern determines the placement of transistors, interconnects, and other components.

Chip Production

What is the Chip Production and Product Transformation Process?

2

Transforming Silicon into a Circuit and Packaging by Using Various Light Sources and Chemical Processes



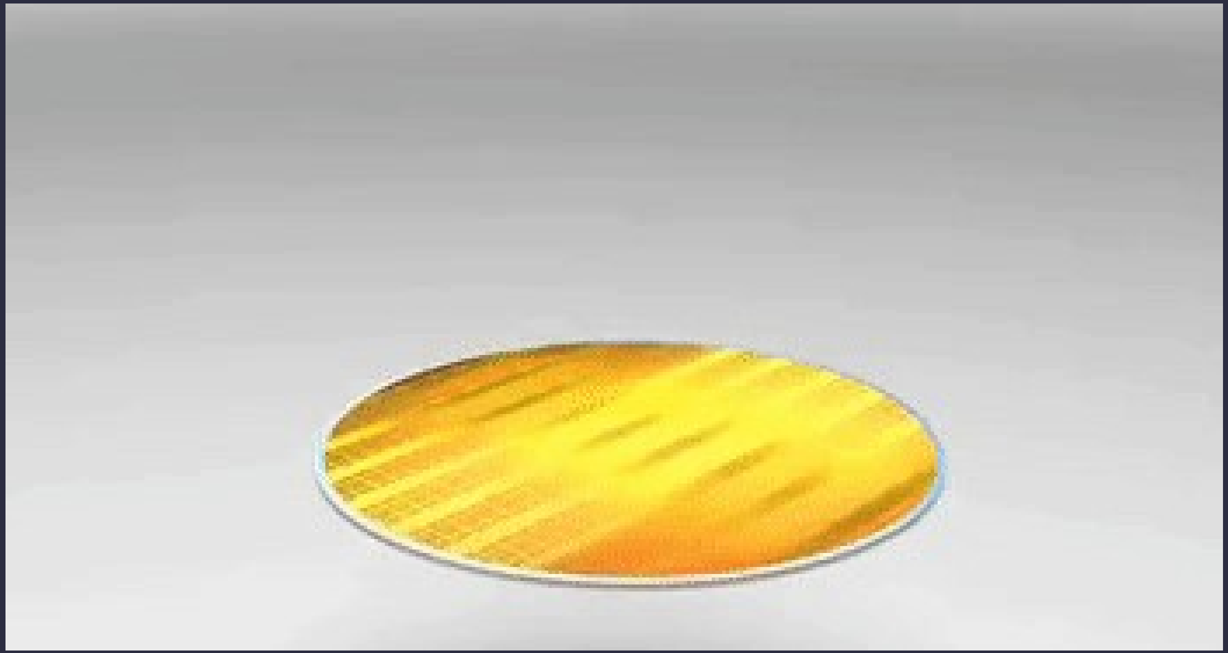
According to the Chip Design, transistors, capacitors, resistors and other components are integrated into the corresponding layers. This process is carried out using photolithography and chemical processes.

Chip Production

What is the Chip Production and Product Transformation Process?

2

Transforming Silicon into a Circuit and Packaging by Using Various Light Sources and Chemical Processes



There can be 10s/100s of different chips on a wafer.
The wafer is cut, the chips are taken, packaged and prepared for use.

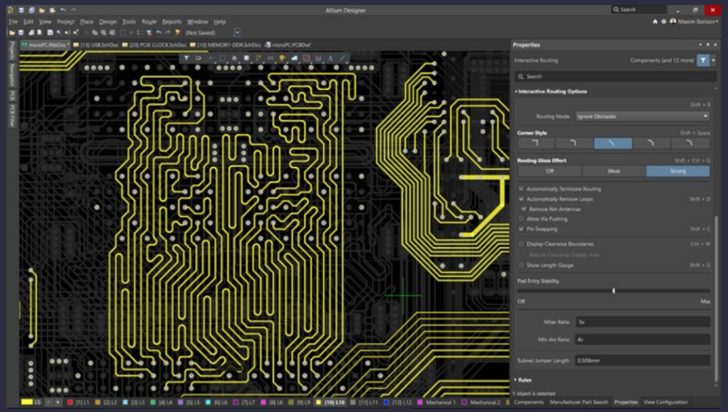
Chip Production

What is the Chip Production and Product Transformation Process?

3

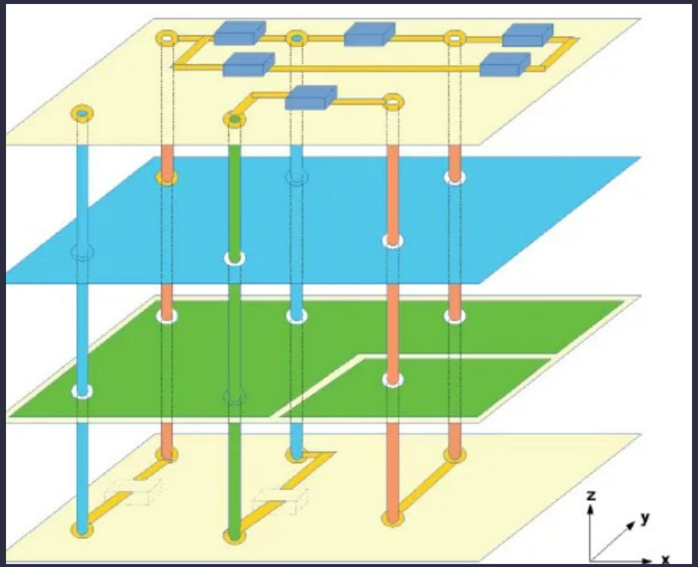
Design of Conductive Copper Sheet (Printed Circuit Board, PCB) to Allow Chips to Transmit Signals to Each Other in a Computer Environment

PCB Design Tools



Multi-Layer Design

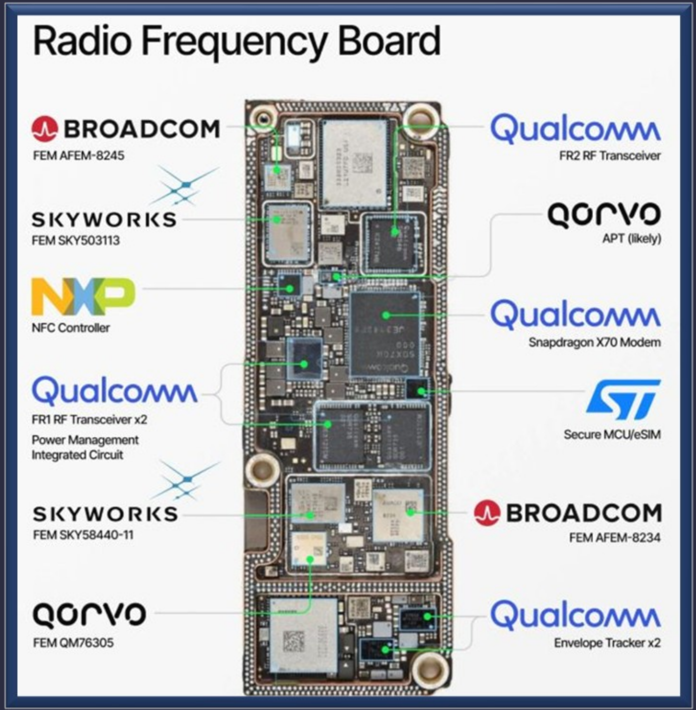
Analysis of the distance signals will travel



Chip Production

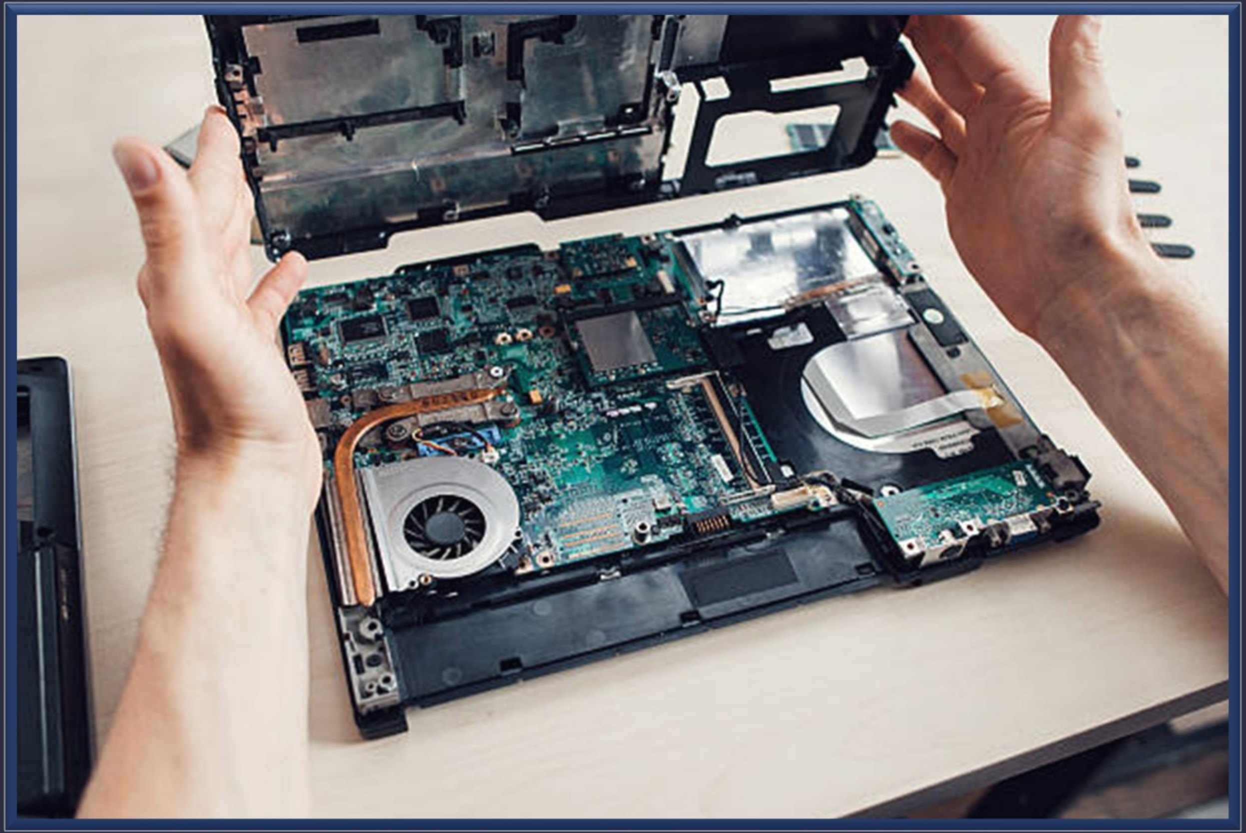
What is the Chip Production and Product Transformation Process?

Iphone 16 Pro



Chip Production

What is the Chip Production and Product Transformation Process?



Laptop PCB

PC Mainboard PCB

Course Plan

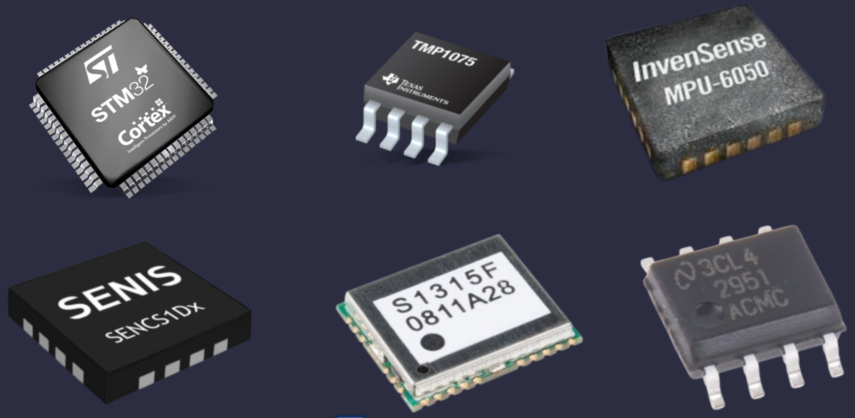
- Introduction
  - Embedded Systems and Applications

Where to start?

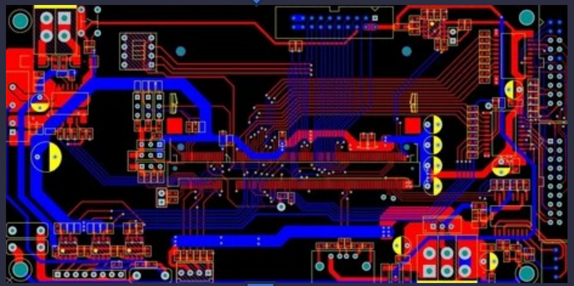
- Buy or Design Required Hardwares
  - Controller
  - Sensors
  - Communication Units
  - Motors
  - Power Regulators
  - Etc..

- Consider constraints…
  - Price
  - Development Time
  - Application Required Standards
  - Availability

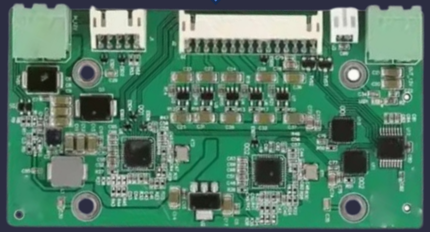
Select Component Supply Or Manufacture



Design and Manufacture PCB



Placement



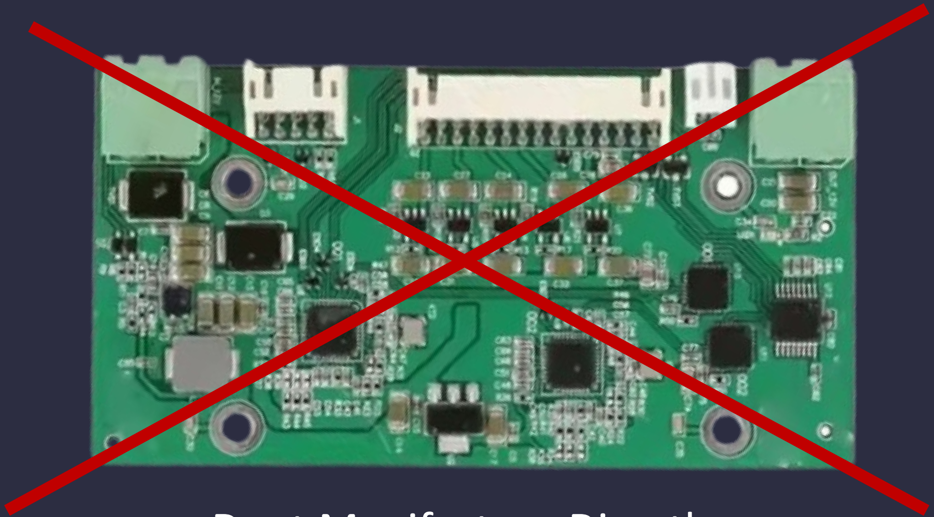
But…

Course Plan

- Introduction
  - Embedded Systems and Applications

Where to start?

- It is not right to buy so many parts and make a PCB without testing whether the system is designed correctly.
- The parts received may be damaged in the slightest mistake.
- There may be something wrongly designed and thought out.
- This will cause loss of time and money.

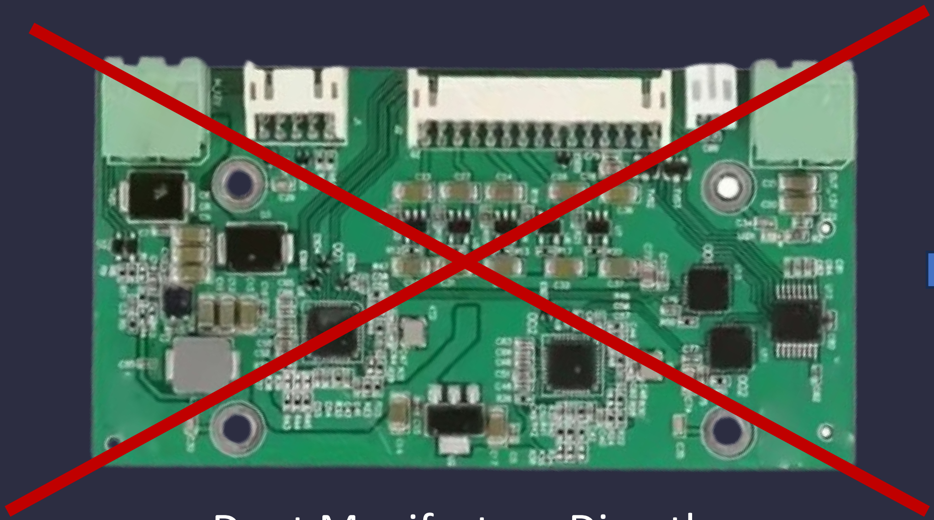


Dont Manifacture Directly

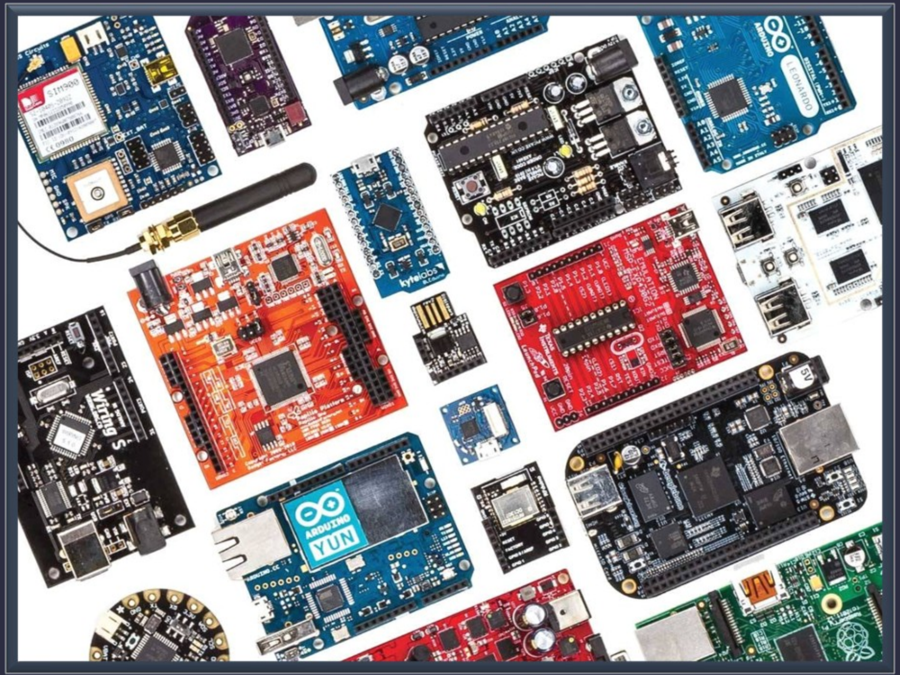
Course Plan

- Introduction
  - Embedded Systems and Applications

Where to start?



Dont Manifacture Directly



Use Development Boards

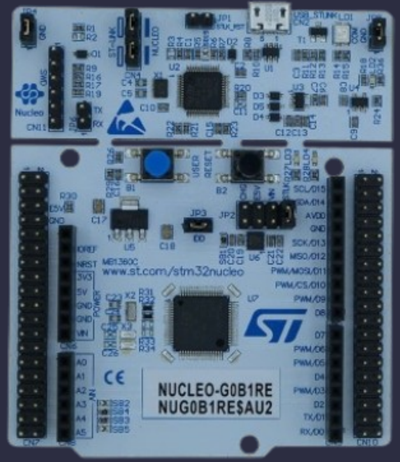
Course Plan

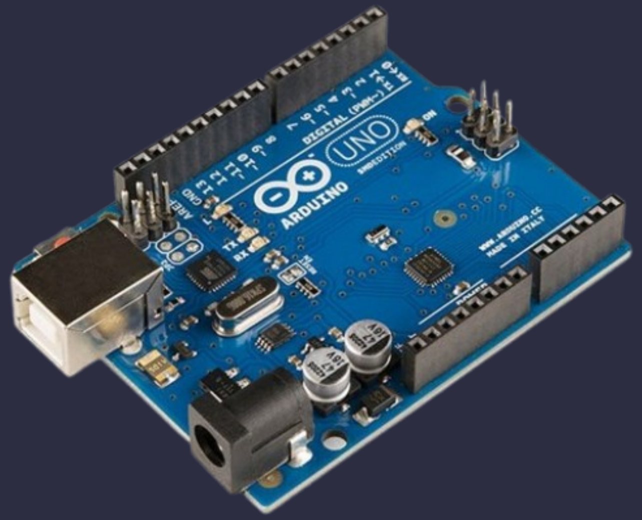
- Introduction
  - Embedded Systems and Applications

Where to start?

Commonly Using MCU Development Boards

- Arduino Uno (Atmega328)
- STM32 Nucleo (STM32U031)



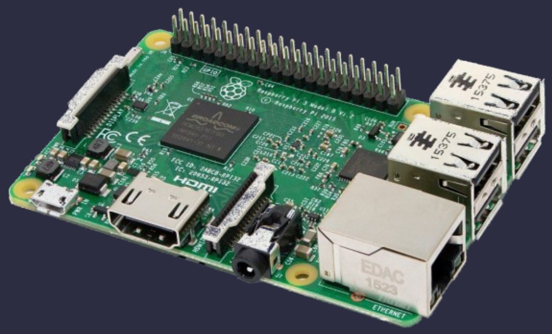


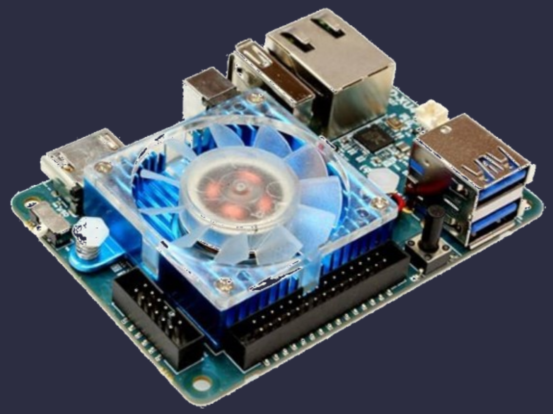
Course Plan

- Introduction
  - Embedded Systems and Applications

Where to start?

Commonly Using MPU Development Boards





- Rasberry Pi 3 (Broadcom BCM2837)
- Odroid xu4 (Samsung Exynos5422)

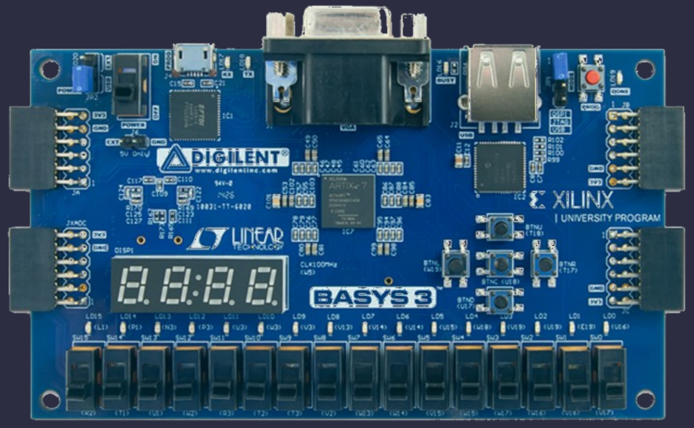
Course Plan

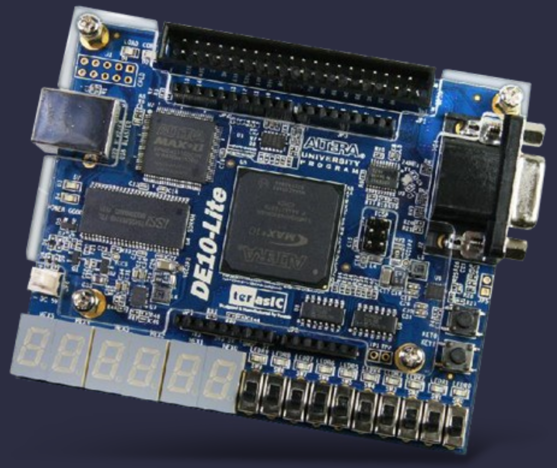
- Introduction
  - Embedded Systems and Applications

Where to start?

Commonly Using FPGA Development Boards

- Basys3 (AMD Artix 7)
- De10-Lite (Intel Max 10)



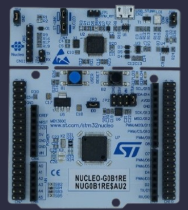


Course Plan

- Introduction
  - Embedded Systems and Applications

What we will do?

Arduino Uno

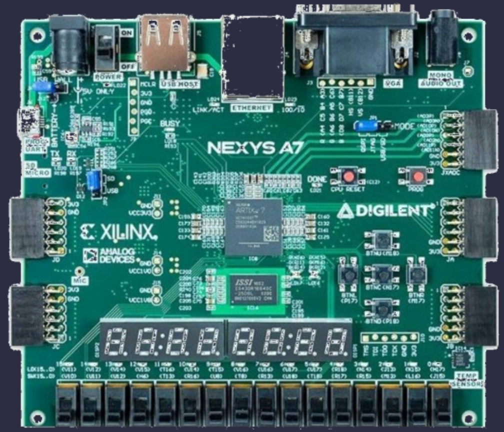


STM32 Nucleo

MCU Based Applications
- GPS
- Temperature
- Pressure
- IMU
- Lorawan

Rasberry Pi 3

MPU Based Applications
- GPS
- Temperature
- Pressure
- IMU
- Lorawan



Nexys A7

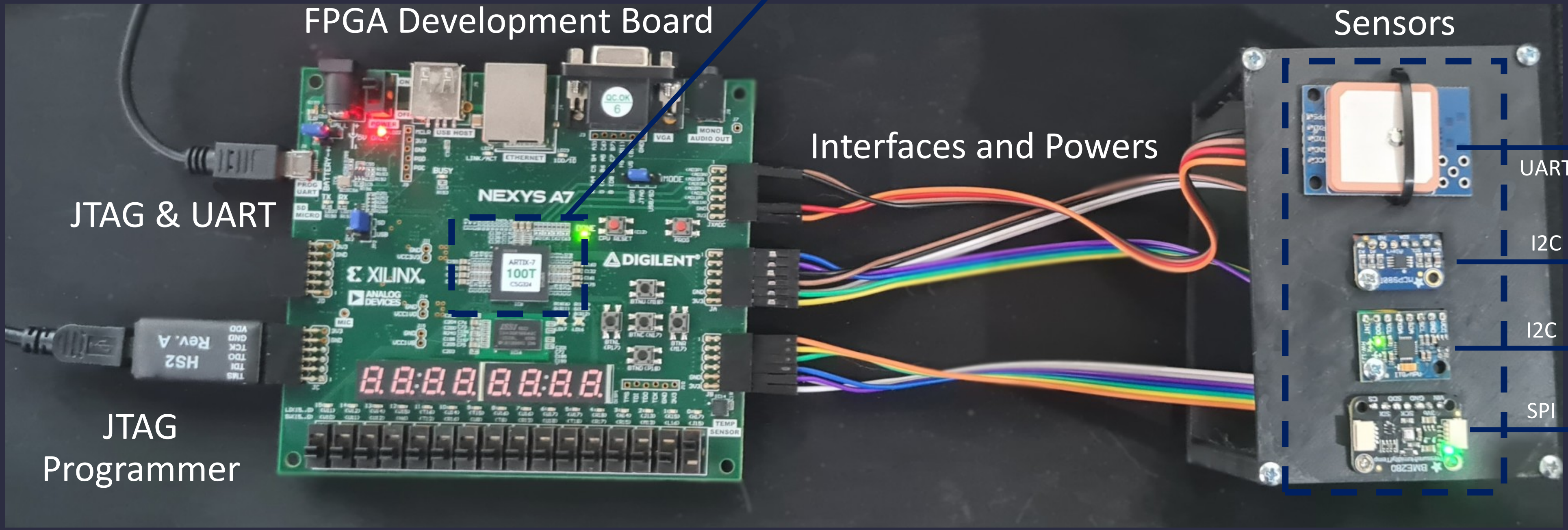
FPGA Based Applications (Design own RISC-V MCU)
- GPS
- Temperature
- Pressure
- IMU
- Lorawan

Course Plan

- Introduction
  - Embedded Systems and Applications

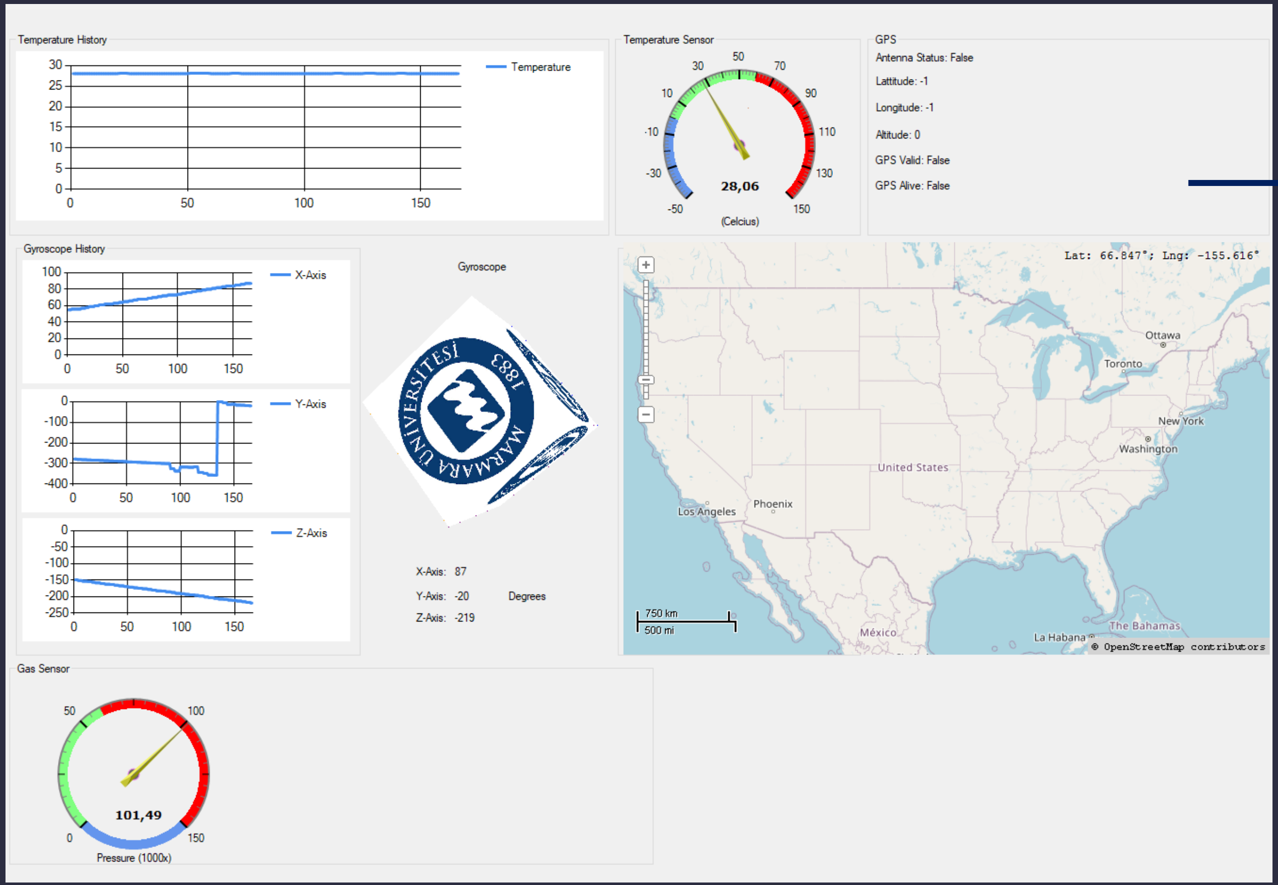
FPGA (AMD Artix 7)

Will contain our own RISC-V MCU



Course Plan

- Introduction
  - Embedded Systems and Applications



Groundstation Software

Course Plan

- Embedded Systems Course Topics

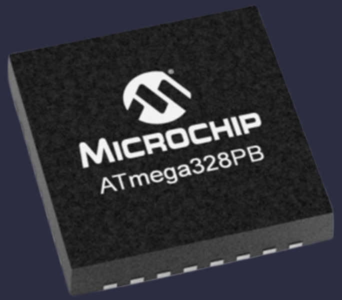
  - Introduction
  - MCU, MPU, FPGA Architectures, Interfaces
  - System Design with Sensors I, Baremetal
  - System Design with Sensors II, FreeRTOS
  - System Design with Sensors III, Embedded Linux
  - RTL Design I (Combinational Circuits)
  - RTL Design II (Sequential Circuits, State Machines)
  - RTL Design III (Memories, Multi-Clock Designs and SoC Concepts) + Basic Verification
  - AXI Interfaces and Custom Hardware Development with AXI Interfaces
  - System Verilog Based Verification
  - RISC-V Architecture and CV32E40P MCU Core
  - RISC-V Baremetal Compilers and FreeRTOS
  - Peripherals (GPIO, UART, Timer, I2C, QSPI, JTAG) and Memory Configurations I + Sensors
  - Peripherals (GPIO, UART, Timer, I2C, QSPI, JTAG) and Memory Configurations II + Sensors

Course Plan

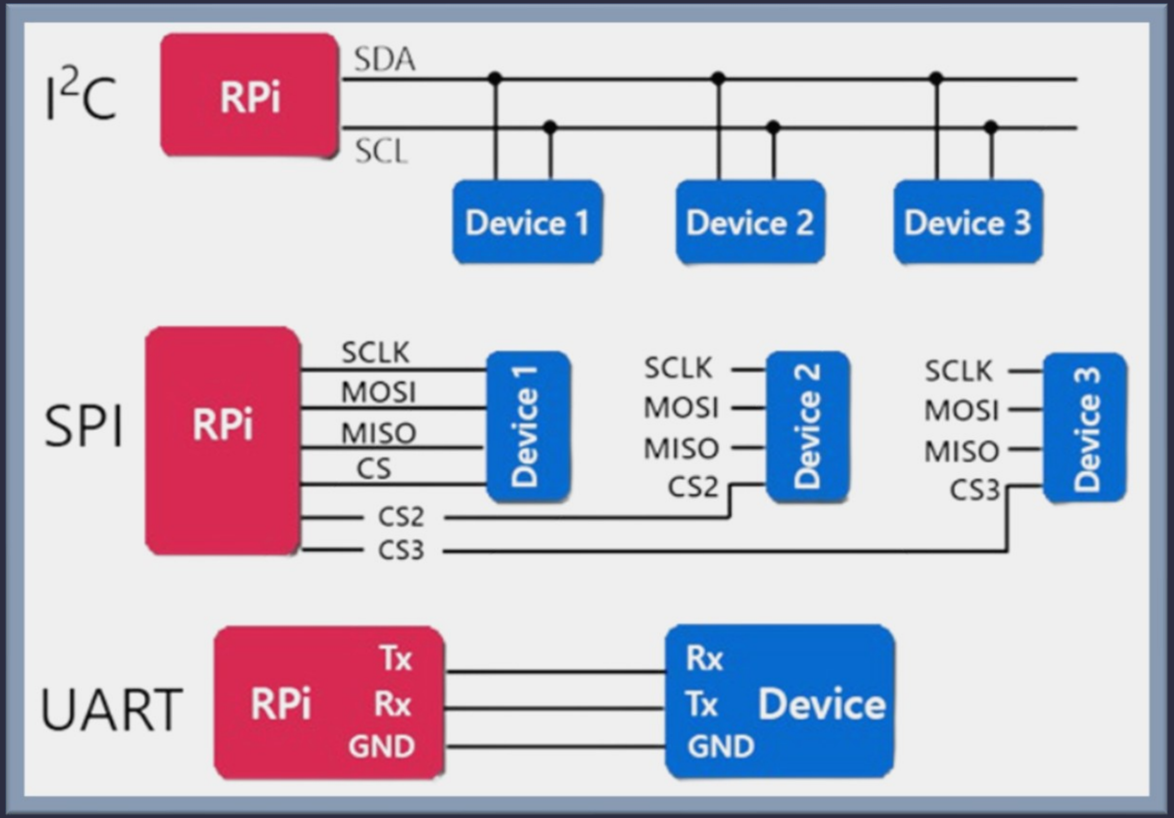
- Embedded Systems Course Topics
  - Introduction

Course Plan

- Embedded Systems Course Topics
  - MCU, MPU, FPGA Architectures, Interfaces

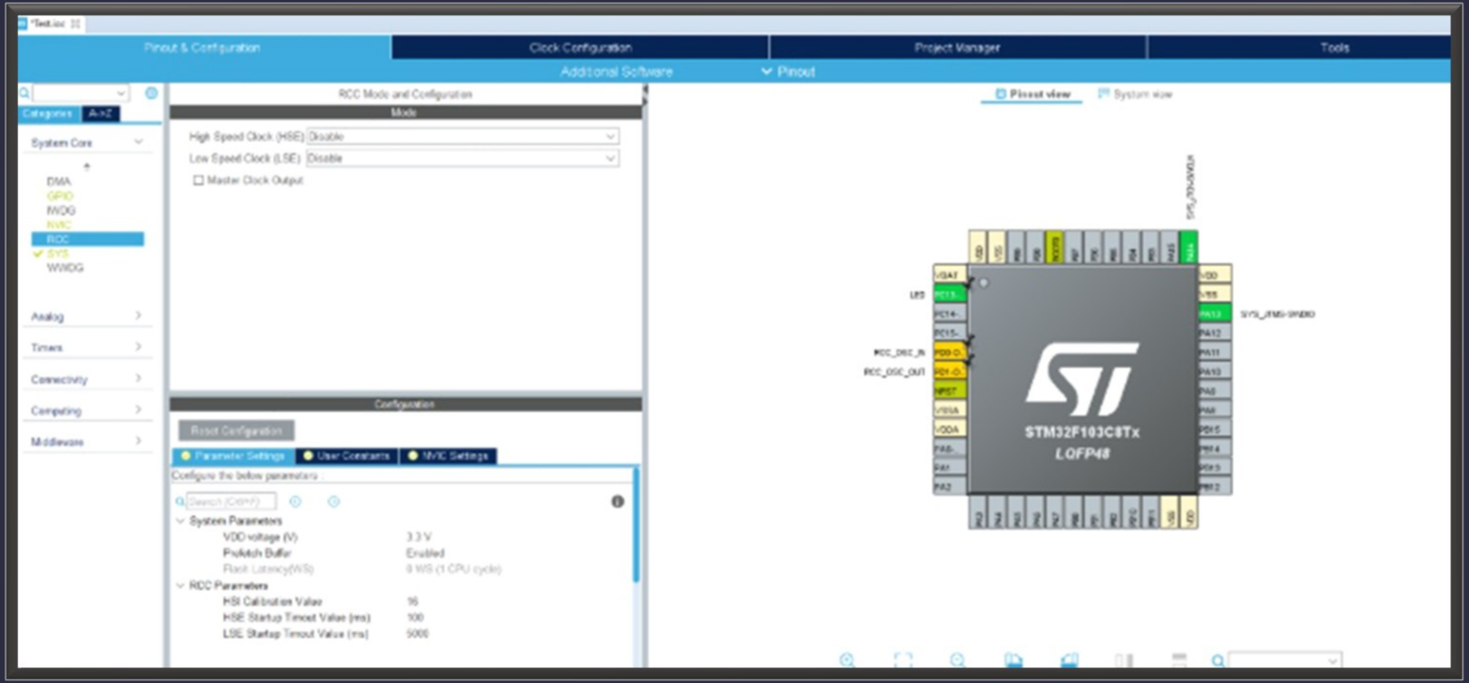


Interfaces



Course Plan

- Embedded Systems Course Topics
  - System Design with Sensors I, Baremetal

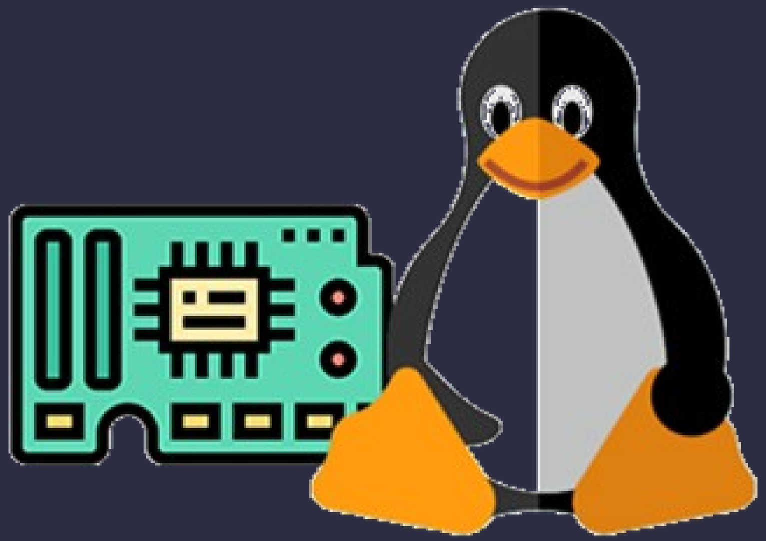


Course Plan

- Embedded Systems Course Topics
  - System Design with Sensors II, FreeRTOS

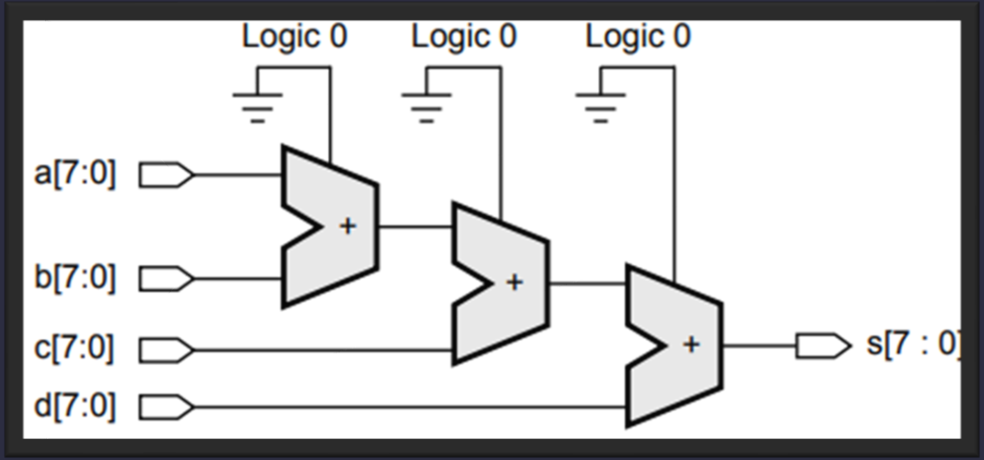
Course Plan

- Embedded Systems Course Topics
  - System Design with Sensors III, Embedded Linux



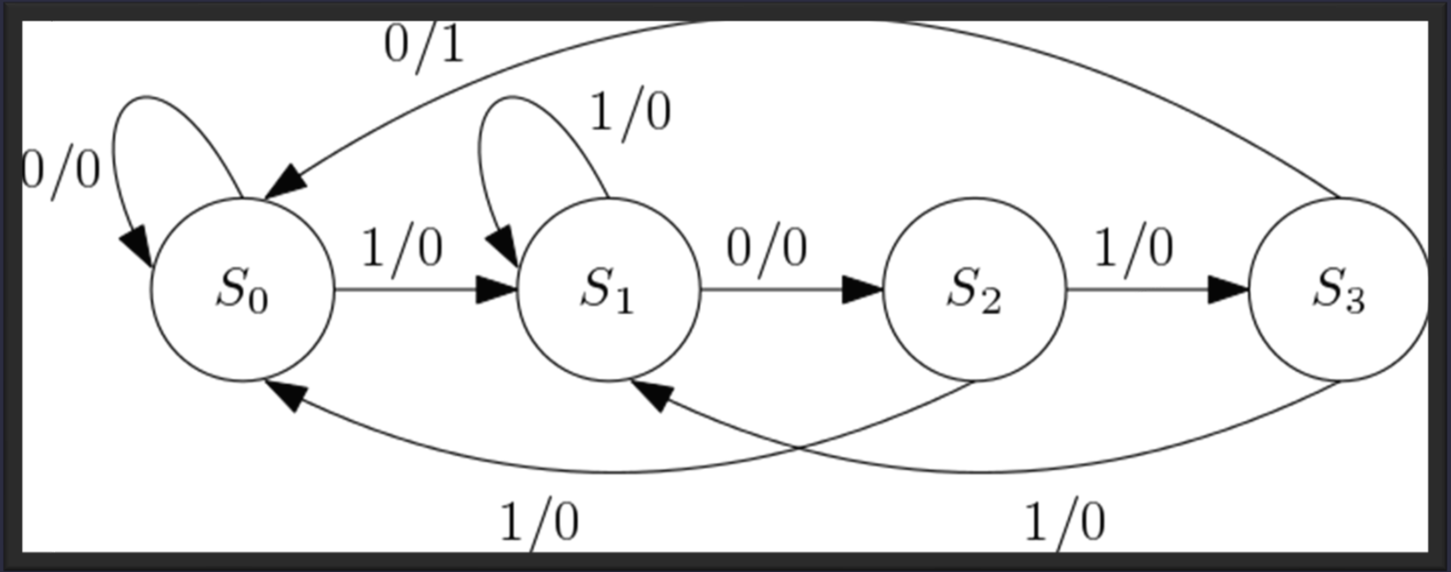
Course Plan

- Embedded Systems Course Topics
  - RTL Design I (Combinational Circuits)



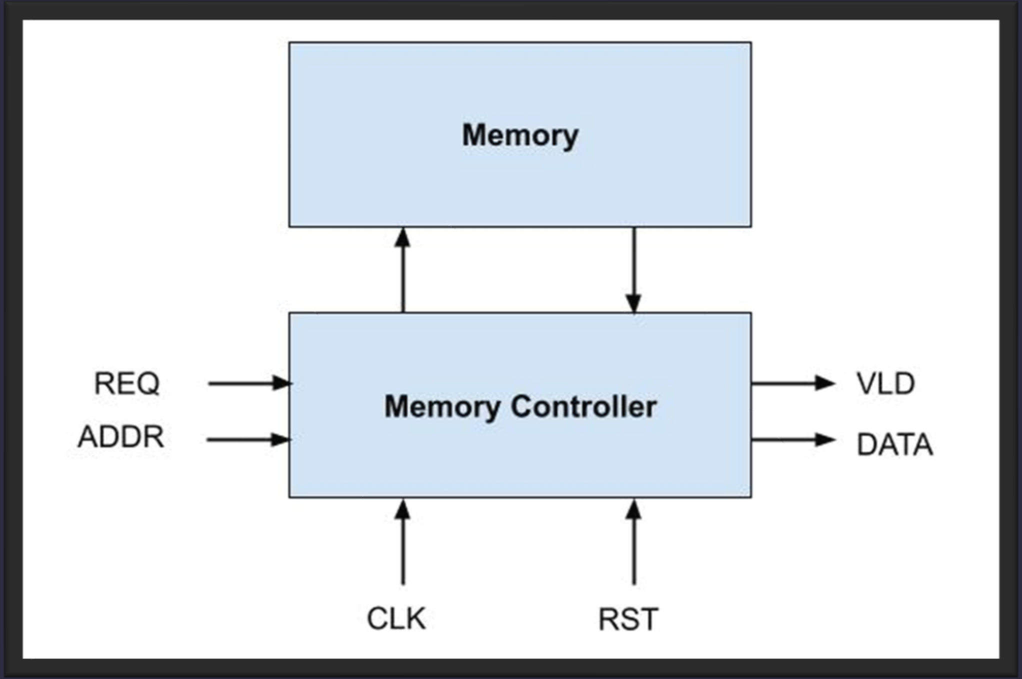
Course Plan

- Embedded Systems Course Topics
  - RTL Design II (Sequential Circuits, State Machines)



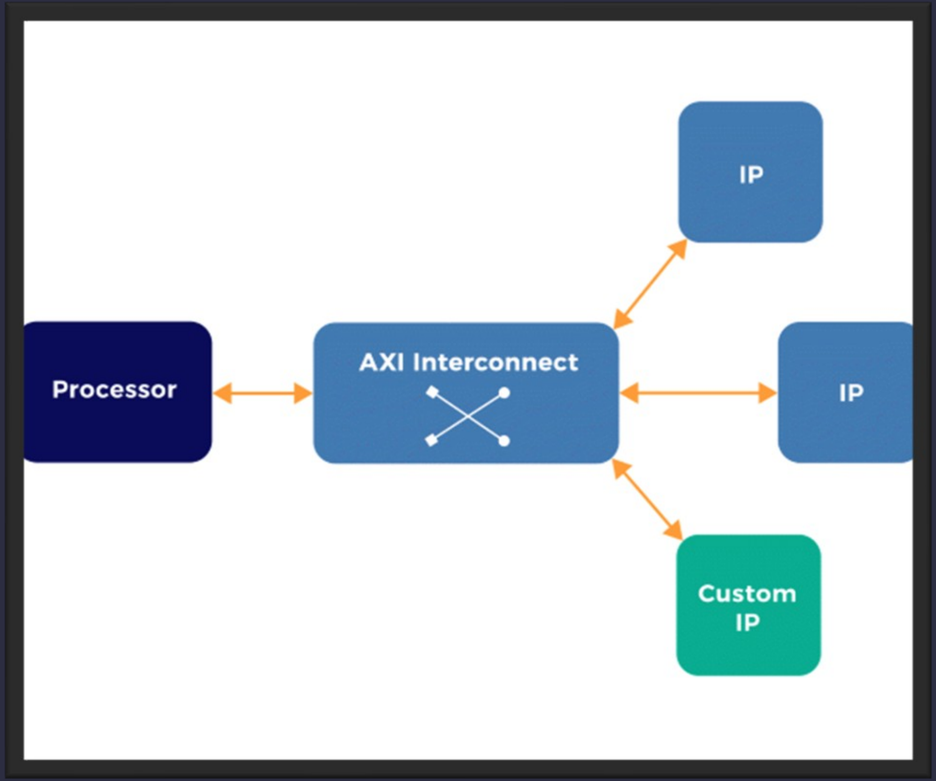
Course Plan

- Embedded Systems Course Topics
  - RTL Design III (Memories, Multi-Clock Designs and SoC Concepts) + Basic Verification



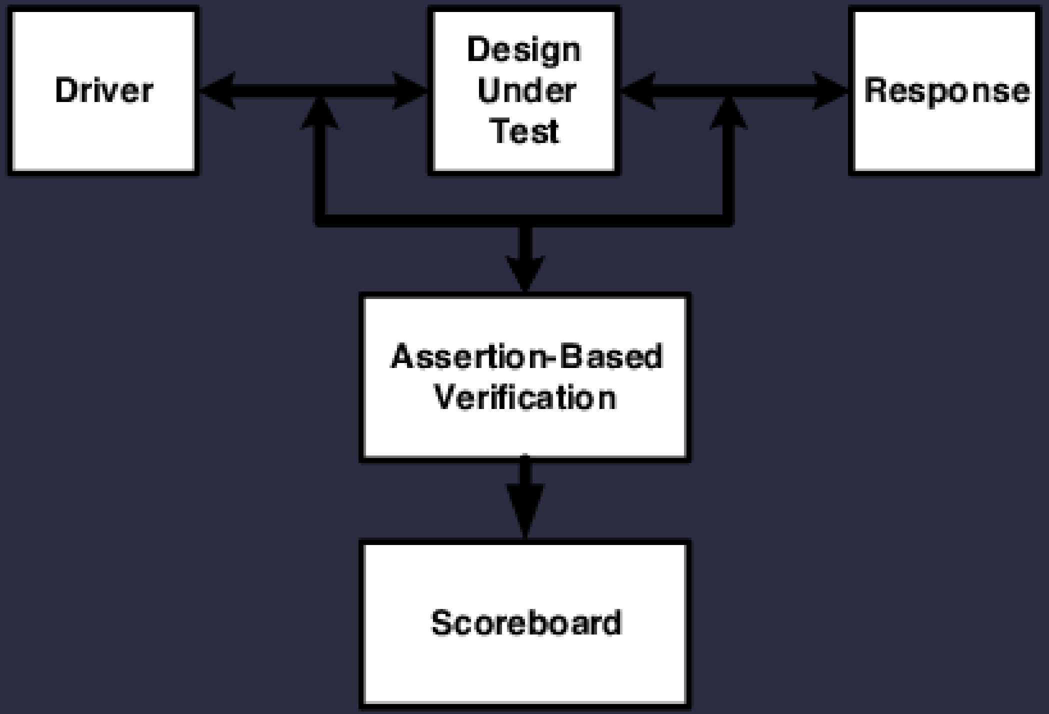
Course Plan

- Embedded Systems Course Topics
  - AXI Interfaces and Custom Hardware Development with AXI Interfaces



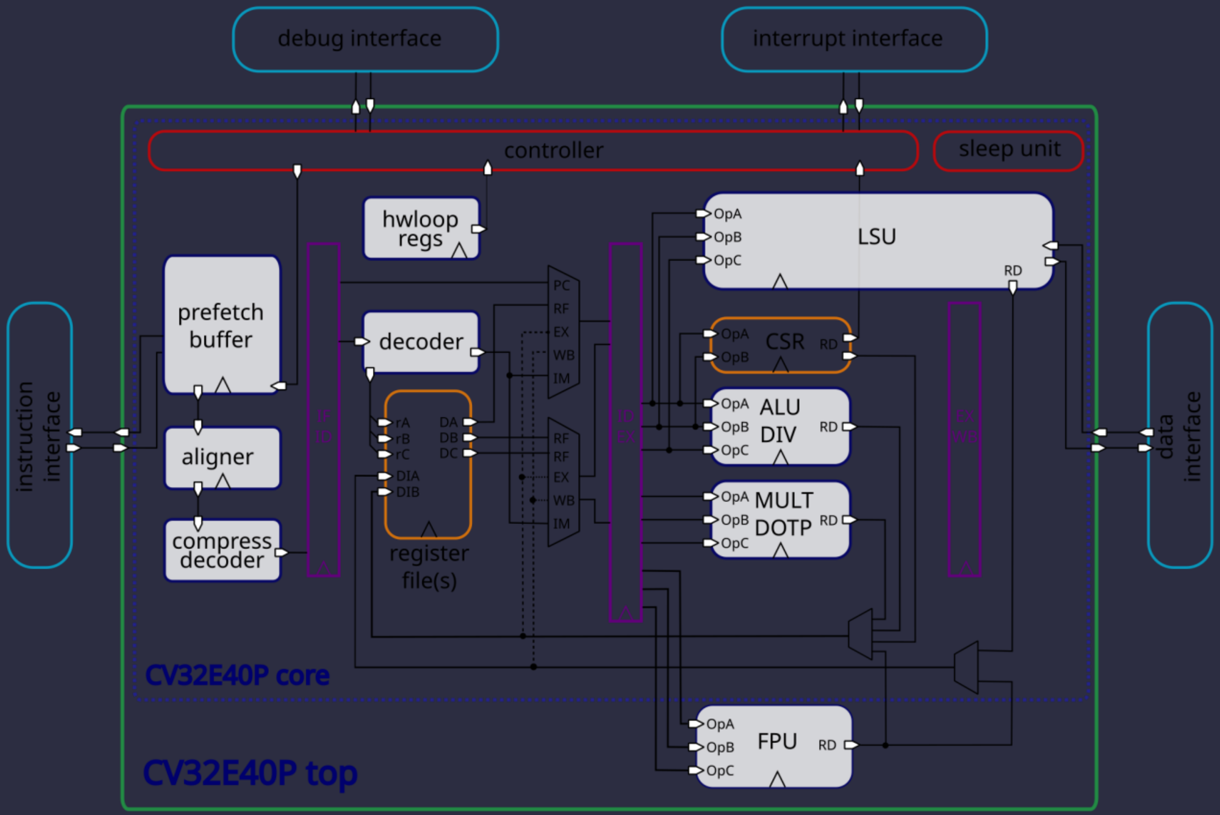
Course Plan

- Embedded Systems Course Topics
  - System Verilog Based Verification



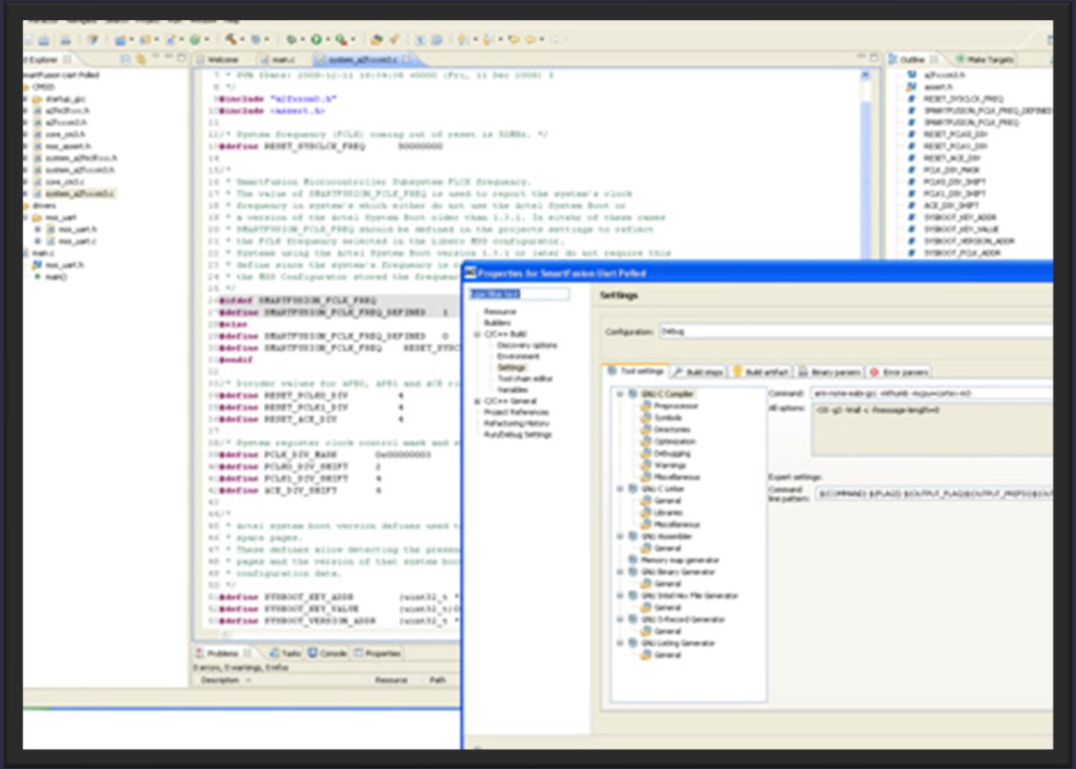
Course Plan

- Embedded Systems Course Topics
  - RISC-V Architecture and CV32E40P MCU Core



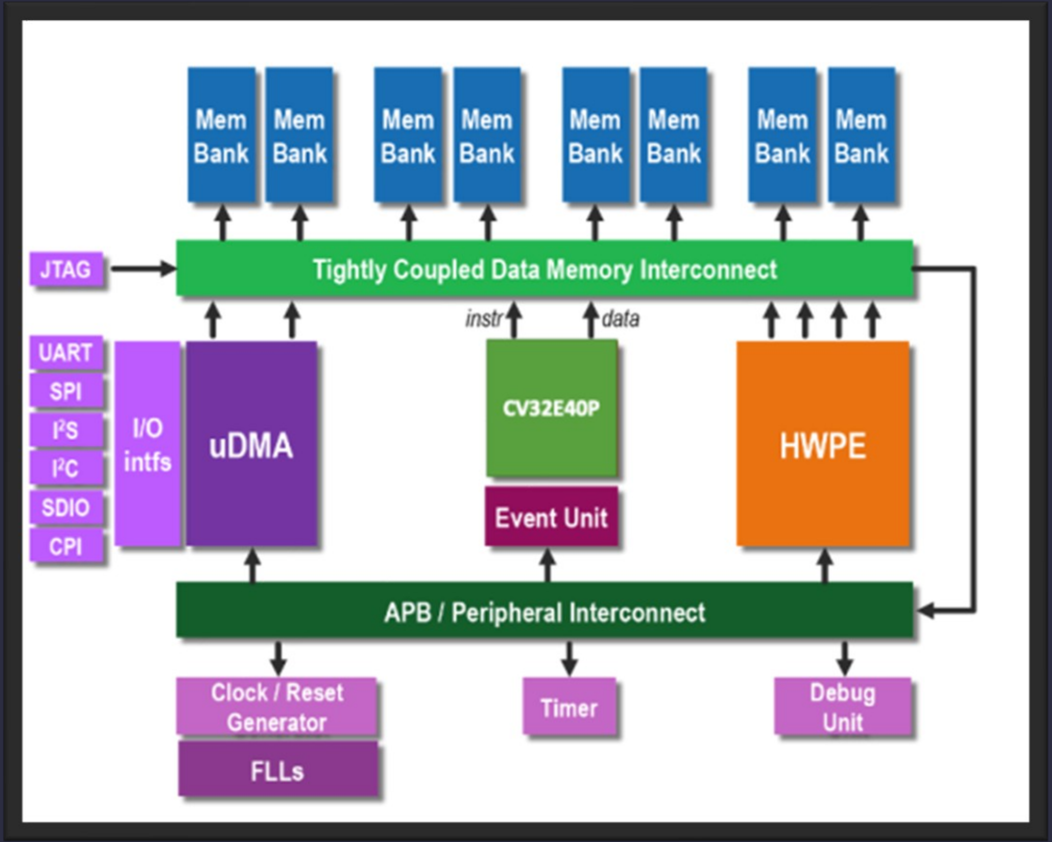
Course Plan

- Embedded Systems Course Topics
  - RISC-V Baremetal Compilers and FreeRTOS



Course Plan

- Embedded Systems Course Topics
  - Peripherals (GPIO, UART, Timer, I2C, QSPI, JTAG) and Memory Configurations + Sensors

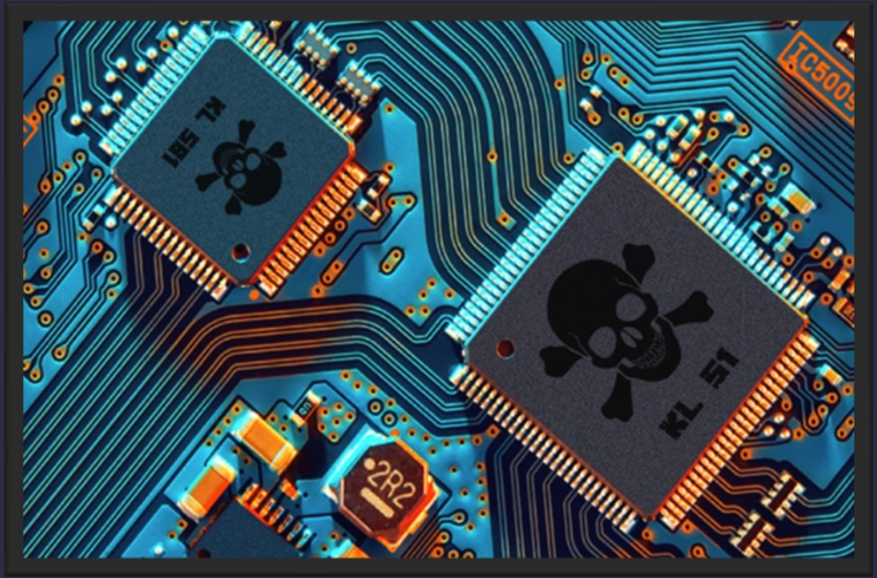


Course Plan

- Introduction
  - Embedded Systems and Applications
  - If you want to a Safety Critical System
    - Automotive
    - Defense
    - Space
  - Very hard to supply electronic components.
    - Like USA ITAR (International Traffic in Arms Regulations) Regulations
    - So you need to implement it!

Course Plan

- Introduction
  - Embedded Systems and Applications
  - Hardware Security
  - Unknown Content in ICs Without In-House Manufacturing
    - Masked production processes can conceal added components.
    - Reverse engineering may not reveal hidden malicious modifications.
  - Risk of Design Theft
    - Third-party manufacturers may copy or alter the design.
    - Unauthorized modifications can introduce vulnerabilities.
  - Types of Hardware Trojans
    - Trigger-Based: Activated externally (signal input) or internally (specific conditions).
    - Effect-Based: Functional modifications, data leakage, or power consumption anomalies.



Hardware Trojans

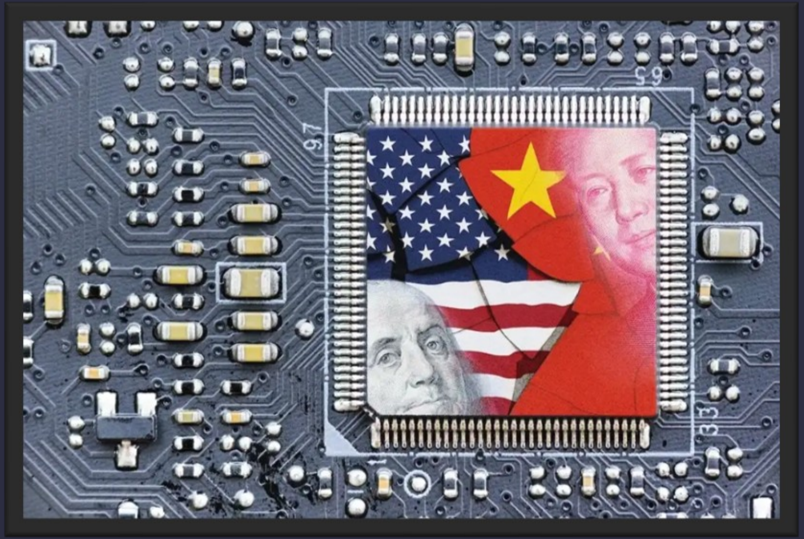
Course Plan

- Introduction
  - Embedded Systems and Applications
  - Chip Manifacturing Crisis
  - Global Semiconductor Shortage
    - Increased demand for electronics during the pandemic.
    - Supply chain disruptions due to factory shutdowns.
    - Rising geopolitical tensions affecting semiconductor trade.
    - Key Causes
      - COVID-19 Pandemic: Production delays and logistics issues.
      - High Demand: Growth in AI, IoT, and 5G technologies.
      - Raw Material Shortages: Limited supply of silicon and rare earth elements.
      - Manufacturing Constraints: Limited foundry capacity at TSMC, Samsung, etc.
      - Geopolitical Factors: US-China trade war and sanctions on key suppliers.
  - Solutions and Future Outlook
    - New Foundries: Intel, TSMC, and Samsung investing in new fabs.
    - Diversification: Companies seeking alternative suppliers and regions.
    - Government Policies: US and EU investing in semiconductor independence.
    - Long-Term Impact: Push for local production and reduced reliance on Asia.

Chip Crisis

Course Plan

- Introduction
  - Embedded Systems and Applications
  - U.S. AI Chip Embargo: Impact on China, Türkiye, and Beyond
  - Reasons for AI Chip Restrictions
    - Preventing military and surveillance applications.
    - Controlling access to cutting-edge AI hardware.
    - Maintaining technological dominance.
    - Countries Affected China: Limited access to AI chips (e.g., NVIDIA A100, H100), pushing local semiconductor development.
    - Türkiye & Others: Restrictions on AI hardware exports due to geopolitical concerns and alliances.
    - Global Impact: Disruptions in AI research, slower tech growth in restricted regions.



AI Chip Embargo

Course Plan

- Introduction
  - Embedded Systems and Applications

  - Opensource CPU Core: RISC-V

  - Open-Source CPU Architecture
    - Free and open instruction set architecture (ISA).
    - Competes with ARM and x86 with no licensing fees.
  - Key Applications
    - Embedded Systems: Low-power IoT, microcontrollers, and industrial automation.
    - AI & HPC: Custom AI accelerators and high-performance computing.
    - Automotive: ADAS and in-vehicle computing.
    - Aerospace & Defense: Secure and customizable processor designs.
  - Major Developments
    - SiFive: Leading RISC-V chip developer with commercial CPU cores.
    - NVIDIA & Intel: Exploring RISC-V for AI and data center applications.
    - China's Push: Developing RISC-V to reduce dependence on Western chips.
  - Challenges & Future Outlook
    - Software Ecosystem: Still maturing compared to ARM/x86.
    - Adoption Growth: Increasing industry support and new chip designs.
    - Geopolitical Influence: Used as an alternative in restricted markets.

Course Plan

- Introduction
  - Embedded Systems and Applications
  - Opensource CPU Core: RISC-V
  - Companies Using RISC-V for AI and Their Work

Tenstorrent
    Develops high-performance, energy-efficient AI processors.
    Licenses RISC-V-based IP cores for AI acceleration.

SiFive
    Leading commercial provider of RISC-V processors.
    Offers customizable RISC-V IP solutions for AI and ML applications.

SpacemiT
    Develops RISC-V AI CPUs and server processors.
    Released Muse Book (RISC-V laptop with Bianbu OS) and announced VitalStone V100 (64-core RISC-V server CPU for 2025).

GreenWaves Technologies
    Produces GAP8, a low-power AI processor optimized for IoT and edge computing.
    Specializes in energy-efficient AI inference at the edge.

Codasip
    Designs RISC-V processor IP for AI and ML workloads
    Provides Codasip Studio, a toolset for custom RISC-V processor design.

Course Plan

- Introduction
  - Embedded Systems and Applications
  - Opensource CPU Core: RISC-V
  - RISC-V AI SoCs Market
    - Expected to grow at a 73.6% CAGR by 2027.
    - AI SoC revenue projected to reach $291 billion by 2027.(Source: Semico Research)
  - Market Size & Growth
    - Semiconductor market valued at $600 billion+ in 2023.
    - Expected to reach $1 trillion by 2030, driven by AI, 5G, and IoT.(Source: McKinsey, Gartner)

Course Plan

- Introduction
  - Embedded Systems and Applications

  - Average Digital Design / RISC-V Hardware Development Engineer Sallary

  - USA, Midlevel 150-200K $ per year
  - TR, Midlevel 150-200K TL per month

  - Over 35K jobs available around the world.

Course Plan

- Introduction
  - Embedded Systems and Applications

Teknofest Chip Design Contest

Link: https://www.teknofest.org/tr/yarismalar/cip-tasarim-yarismasi/

Expected Output: RISC-V Based MCU with UART, SPI, I2C and Verification

Course Plan

- Introduction
  - Embedded Systems and Applications

Component Suppliers

Components

- Digikey
- Mouser
- Farnell

PCB

- JLCPCB
- PCBWAY

Chip Manifacturing

- Efabless OpenLane (Free or 10K$ for 200px)

For Turkey

- Ekom
- Ozdisan
- Elektrovadi
- DirencNet
- Robotistan

Course Resources

Website: levent.tc

Courses > Graduate Courses -> Embedded Systems[Grad-TR] (Turkish)

Course Resources

Course Page Content;

- Detailed Course Description
- Syllabus
- Lecture Notes
- Homeworks
- Research Topics
- Exams and Sample Questions
- LMS

Course Resources

Syllabus;

Lesson hours;

Monday 13.00-16.00

Course Resources

Syllabus;

Between 4-6 homeworks will be given.

Attendance to classes is mandatory at **80** %.

Course Resources

Syllabus;

Evaluation weights

Activities	Rates
Midterm	20%
Homework/Quiz	30%
Project	50%

Deadline for homework for every passing hour 5 points will be deducted.

Course Resources

Syllabus;

Letter grade ranges

Term Grade	Weight	Letter grade
90-100	4.00	AA
85-89	3.50	BA
80-84	3.00	BB
75-79	2.50	CB
65-74	2.00	CC
50-64	1:50	DC
45-49	1.00	DD
0 -44	0	FF

Course Resources

Syllabus;

expected effort

180 hours in total effort is expected.

Contents	Hour	How many times	Subtotal
Lesson Preparation	2	15	30
Lesson Repetition	2	15	30
Homework	4	6	24
Project	48	1	48
Classroom Course	3	15	45
Midterm	3	1	3

Course Resources

Syllabus;

Academic honesty

Course Resources

Course Schedule;

Week	Subject
1	Introduction
2	MCU, MPU, FPGA Architectures, Interfaces
3	System Design with Sensors I, Baremetal
4	System Design with Sensors II, FreeRTOS
5	System Design with Sensors III, Embedded Linux
6	RTL Design I (Combinational Circuits)
7	RTL Design II (Sequential Circuits, State Machines)
8	RTL Design III (Memories, Multi-Clock Designs and SoC Concepts) + Basic Verification
9	**Midterm**
10	AXI Interfaces and Custom Hardware Development with AXI Interfaces
11	System Verilog Based Verification
12	RISC-V Architecture and CV32E40P MCU Core
13	RISC-V Baremetal Compilers and FreeRTOS
14	Peripherals (GPIO, UART, Timer, I2C, QSPI, JTAG) and Memory Configurations I + Sensors
15	Peripherals (GPIO, UART, Timer, I2C, QSPI, JTAG) and Memory Configurations II + Sensors

Course Resources

Homeworks;

Assignments to be given and their solutions will be shared on the homework page.

Course Resources

Projects;

At the end of the semester, the projects that each student should do will be announced.

Course Resources

Exams;

Solutions of sample questions and exams will be shared for midterm

Course Resources

LMS;

The Blackboard LMS system is the system where we will ask you to upload some of the assignments that will be given to you. The system will automatically shut down on the last upload date.